



**UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE CIÊNCIAS EXATAS E NATURAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO**

ALLAN DOUGLAS BENTO DA COSTA

**ALGORITMO DE GESTÃO DE MOBILIDADE PARA
DISSEMINAÇÃO DE VÍDEOS EM REDES VEICULARES
ULTRADENSAS**

Belém
2020



UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE CIÊNCIAS EXATAS E NATURAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

ALLAN DOUGLAS BENTO DA COSTA

ALGORITMO DE GESTÃO DE MOBILIDADE PARA
DISSEMINAÇÃO DE VÍDEOS EM REDES VEICULARES
ULTRADENSAS

Tese apresentada ao Programa de Pós-Graduação em Ciência da Computação. Instituto de Ciências Exatas e Naturais. Universidade Federal do Pará como parte dos requisitos para a obtenção do título de Doutor em Ciência da Computação.

Área de Concentração: Sistemas de Computação
Orientador: Prof. Dr. Eduardo Coelho Cerqueira
Coorientador: Prof. Dr. Denis Lima do Rosário

Belém
2020

**Dados Internacionais de Catalogação na Publicação (CIP) de acordo com ISBD
Sistema de Bibliotecas da Universidade Federal do Pará
Gerada automaticamente pelo módulo Ficat, mediante os dados fornecidos pelo(a) autor(a)**

C837a Costa, Allan Douglas Bento
ALGORITMO DE GESTÃO DE MOBILIDADE PARA
DISSEMINAÇÃO DE VÍDEOS EM REDES VEICULARES
ULTRADENSAS / Allan Douglas Bento Costa. — 2020.
102 f. : il. color.

Orientador(a): Prof. Dr. Eduardo Coelho Cerqueira
Coorientador(a): Prof. Dr. Denis Lima do Rosário
Tese (Doutorado) - Programa de Pós-Graduação em Ciência da
Computação, Instituto de Ciências Exatas e Naturais, Universidade
Federal do Pará, Belém, 2020.

1. VANETs. 2. Handover. 3. Predição de Mobilidade. 4.
Reconhecimento de QoE. 5. Redes Ultradensas. I. Título.

CDD 004



UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE CIÊNCIAS EXATAS E NATURAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

ALLAN DOUGLAS BENTO DA COSTA

**ALGORITMO DE GESTÃO DE MOBILIDADE PARA DISSEMINAÇÃO DE
VÍDEOS EM REDES VEICULARES ULTRADENSAS**

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal do Pará como requisito para obtenção do Título de Doutor em Ciência da Computação, defendida e aprovada em 10/06/2020, pela banca examinadora constituída pelos seguintes membros:

Eduardo Coelho Cerqueira

Prof. Dr. Eduardo Coelho Cerqueira
Orientador – PPGCC/UFPA

Denis Lima do Rosário

Prof. Dr. Denis Lima do Rosário
Co-orientador – PPGCC/UFPA

André Figueira Riker

Prof. Dr. André Figueira Riker
Membro Externo – FACOMP/UFPA

Helder May Nunes da Silva Oliveira

Prof. Dr. Helder May Nunes da Silva Oliveira
Membro Externo – FACOMP/UFPA

Thaís Lira Tavares dos Santos

Profa. Dra. Thaís Lira Tavares dos Santos
Membro Externo – UFPA

Visto: *Nelson Cruz Sampaio Neto.*

Prof. Dr. Nelson Cruz Sampaio Neto
Coordenador do PPGCC/UFPA

AGRADECIMENTOS

Em primeiro lugar, quero dirigir os meus agradecimentos ao meu orientador, Prof. Dr. Eduardo Cerqueira¹ pela orientação prestada e oportunidade de realizar este trabalho, pela confiança e por me atender com paciência. Agradeço por todos seus ensinamentos compartilhados de forma admirável, pelo seu incentivo, disponibilidade, seus preciosos conselhos, sua inestimável confiança, apoio que sempre demonstrou e por me guiar nos primeiros passos do Mestrado ao término deste Doutorado. Aqui lhe exprimo a minha eterna admiração, respeito e gratidão.

Ao meu coorientador Prof. Dr. Denis do Rosário² pelo seu incentivo, pela sua disponibilidade, estímulo ao desenvolvimento desta tese e fundamentais ensinamentos que contribuíram de forma incomensurável ao longo dos anos de pesquisa.

Manifesto aqui a minha gratidão a todos Professores e Funcionários do PPGCC da UFPA, em particular, ao Prof. Dr. Nelson Neto³. Obrigado pela demonstração de sabedoria e humildade.

A todos amigos e companheiros de trabalho do *Research Group on Computer Networks and Communication Multimedia* - (GERCOM)⁴ pelos debates enriquecedores, pelo compartilhamento de experiências de pesquisa e estudo e pelo clima sempre acolhedor.

Meus respeitosos agradecimentos pela contribuição da banca do exame de qualificação e pela participação dos membros da banca examinadora da defesa.

¹ <<http://lattes.cnpq.br/1028151705135221>>

² <<http://lattes.cnpq.br/8273198217435163>>

³ <<http://lattes.cnpq.br/9756167788721062>>

⁴ <<http://www.gercom.ufpa.br/>>

*Não se pode ensinar tudo a alguém, pode-se apenas
ajudá-lo a encontrar por si mesmo o caminho.
(Galileu Galilei)*

RESUMO

Resumo de Tese de Doutorado apresentada à UFPA como parte dos requisitos para obtenção do grau de Doutor em Ciência da Computação pelo Programa de Pós-Graduação em Ciência da Computação.

ALGORITMO DE GESTÃO DE MOBILIDADE PARA DISSEMINAÇÃO DE VÍDEOS EM REDES VEICULARES ULTRADENSAS

Orientador: Prof. Dr. Eduardo Coelho Cerqueira

Coorientador: Prof. Dr. Denis Lima do Rosário

Palavras-chave: VANETs, Handover, Predição de Mobilidade, Predição de QoE, Redes Ultra Densa.

As redes da próxima geração tem como objetivo iniciar a distribuição de vídeo por redes *ad-hoc* veiculares (VANET), as quais são compostas por redes heterogêneas ultradensas. Essas redes leva em conta a infraestrutura de comunicação existente, com o intuito de obter maior eficiência espectral, tal como, a reutilização de taxas do espectro. No entanto, o aumento do número de células acaba por tornar o processo de gerenciamento de mobilidade (ou seja, *handover*) uma tarefa desafiadora em VANETs ultradensas, já os veículos alternam frequentemente entre redes distintas, levando ao aumento de *handover* desnecessário, sobrecarregando a rede. Esta decorrência chamado efeito *ping-pong*, o qual gera inúmeras perdas de e/ou quadros de vídeo, tornando o algoritmo de *handover* por fornecer vídeos de baixa Qualidade de Experiência (QoE). Para contornar esse problema, esta tese de doutorado apresenta um algoritmo eficiente de *handover* para distribuição de conteúdo de vídeo em VANET ultradensa, o qual será chamado de Skip-HoVe. O algoritmo proposto apresenta as seguintes características: um mecanismo de *handover* ciente de predição de mobilidade e QoE, bem como Qualidade de Serviço (QoS), resultando em um *handover* entre as células com maior confiabilidade e menor frequência. Os resultados da simulação mostram a eficiência do algoritmo Skip-HoVe 30% superior comparada a algoritmos de *handover* existentes, mantendo uma taxa de *ping-pong* em torno de 2%.

ABSTRACT

Abstract of Doctoral Thesis presented to UFPA as part of the requirements for obtaining a PhD degree in Computer Science by the Graduate Program in Computer Science.

An Efficient Handover Algorithm for Dissemination of Video Content in Ultra-Dense Vehicle Networks

Advisor: Prof. Dr. Eduardo Coelho Cerqueira

Co-advisor: Prof. Dr. Denis Lima do Rosário

Keywords: VANETs, Handover, Mobility Prediction, QoE Prediction, and ultra-dense network.

The next generation networks aim to pave the way for video distribution via vehicular ad-hoc networks (VANET), which will be composed of heterogeneous ultra-dense networks. These networks will take into account the existing communication infrastructure, in order to obtain greater spectral efficiency, such as the reuse of spectrum fees. However, the increase in the number of cells ends up making the mobility management process (ie, handover) a challenging task in ultra-dense VANETs. This is because vehicles often move between different networks, leading to an increase in unnecessary handover and greater network overhead, called the ping-pong effect. In this sense, an inefficient handover algorithm ends up providing videos with low Quality of Experience (QoE), which is caused by ping-pong handover decisions, which increases the packet and video frames losses. This thesis presents a handover algorithm for video distribution over ultra-dense VANET, called of Skip-HoVe. The proposed algorithm has the following characteristics: a handover mechanism aware of Mobility Prediction and QoE, as well as quality of service, resulting in handover with greater reliability and less frequency. The simulation results show the efficiency of the Skip-HoVe algorithm to deliver videos with 30% higher performance compared to existing handover algorithms, maintaining a ping-pong rate of around 2%.

LISTA DE ILUSTRAÇÕES

Figura 1 – Exemplo de um cenário VANET Ultradensa.	18
Figura 2 – Aplicações de vídeos em tempo real e sob demanda.	23
Figura 3 – Topologia Dinâmica das Redes V2V, V2I, V2P e outras	24
Figura 4 – Tendências no Processo de Handover	25
Figura 5 – Diagrama de Qualidade de Experiência.	26
Figura 6 – Avaliação objetiva de QoE através das métricas NR	29
Figura 7 – Avaliação objetiva de QoE através das métricas RR.	29
Figura 8 – Avaliação objetiva de QoE através das métricas FR.	30
Figura 9 – Etapas de Gerenciamento de <i>Handover</i>	48
Figura 10 – Medição de Previsão de Mobilidade	54
Figura 11 – RMSE para o ARIMA e para o filtro Kalman	57
Figura 12 – As previsões de longitude do veículo para o ARIMA e KF.	58
Figura 13 – Amostra do Cenário de Simulação.	60
Figura 14 – SSIM para vídeos entregues por diferentes algoritmos de <i>handover</i>	63
Figura 15 – MOS para vídeos entregues por diferentes algoritmos de <i>handover</i>	64
Figura 16 – Taxa de perda de Quadros I para vídeos entregues por diferentes algoritmos de <i>handover</i>	64
Figura 17 – Comparativo do 683º quadro do vídeo transmitido por diferentes algoritmos de <i>handover</i>	66
Figura 18 – Valor MOS - Alto	67
Figura 19 – Valor MOS - Baixo	67
Figura 20 – Número de <i>handover</i> executadas por diferentes algoritmos de <i>handover</i>	68
Figura 21 – Taxa de <i>handover</i> com efeito ping-pong por diferentes algoritmos de <i>handover</i>	69

LISTA DE TABELAS

Tabela 1 – Tabela Descritiva do MOS	27
Tabela 2 – Quadro Comparativo de Trabalhos Relacionados	44
Tabela 3 – Importância do Contexto em Pares	51
Tabela 4 – Parâmetros de Simulação Aplicados	59
Tabela 5 – Resumo de resultados para um veículo no cenário.	69

LISTA DE ABREVIATURAS E SIGLAS

AHP	<i>Analytic Hierarchy Process</i>
ARIMA	<i>AutoRegressive Integrated Moving Average</i>
AU	<i>Application Unit</i>
CSH	<i>Service Chanel</i>
CCH	<i>Control Channel</i>
CR	<i>Consistency Ratio</i>
CI	<i>Consistency Index</i>
DSRC	<i>Dedicated Short-Range Communication</i>
DCT	<i>Discrete Cosine Transform</i>
DC	<i>Direct Current</i>
ELP	<i>Electronic License Plate</i>
EPC	<i>Envolved Packet Core</i>
EDR	<i>Event Data Recorder</i>
FAHP	<i>Fuzzy Analytical Hierarchical Process</i>
GPS	<i>Global Positioning System</i>
GoP	<i>Group of Pictures</i>
GI	<i>Graphical Interface</i>
HVS	<i>Human Visual System</i>
H-UDN	<i>Heterogeneous Ultra-Dense Network</i>
ITS	<i>Intelligent Transportation System</i>
KF	<i>Kalman Filter</i>
LTE	<i>Long Term Evolution</i>
LLC	<i>Logic Link Control</i>
MIMO	<i>Multiple-input multiple-output</i>
MM Wave	<i>Millimeter Wave</i>

MOS	<i>Mean Opinion Score</i>
MIMO	<i>Multiple-input multiple-output</i>
MANET	<i>Mobile ad hoc network</i>
MAC	<i>Media Access Control</i>
MIB	<i>Management Information Base</i>
MSE	<i>Mean Squared Error</i>
ML	<i>Machine Learning</i>
MIH	<i>Media Independent Handover</i>
MADM	<i>Multiple Attributes Decision Algorithms</i>
OBU	<i>On-BoardUnit</i>
PoA	<i>Point of Attachment</i>
PSOBUs	<i>Public Safety OBUs</i>
PSNR	<i>Peak Signal to Noise Ratio</i>
PDR	<i>Packet Delivery Rate</i>
QoS	<i>Quality of Service</i>
QoE	<i>Quality of Experience</i>
RSS	<i>Received Signal Strength</i>
RSSI	<i>Received Signal Strength Indicator</i>
RSU	<i>Road Side Units</i>
RMA	<i>Resource Manager Applications</i>
RSRQ	<i>Reference Signal Received Quality</i>
RM	<i>Random Forests</i>
RMSD	<i>Root-Mean-Square Deviation</i>
RLF	<i>Radio Link Failure</i>
Skip-HoVe	<i>Handover HoVe</i>
SINR	<i>Signal to Interference Plus Noise Ratio</i>

SSIM	<i>Structural Similarity Index</i>
SCeNBs	<i>Small Cellbase Stations</i>
TTT	<i>Time To Trigger</i>
VQM	<i>Video Quality Metric</i>
VQA	<i>Video Quality Assessment</i>
VCM	<i>VANET Connection Manager</i>
VF	<i>Video Flow</i>
LC	<i>Local Contrast</i>
VDN	<i>Video Distribution Networks</i>
V2V	<i>Vehicle-to-Vehicle</i>
V2R	<i>Vehicle-to-Roadside</i>
V2I	<i>Vehicle-to-Infrastructure</i>
V2P	<i>Vehicle-to-Pedestrian</i>
V2D	<i>Vehicle-to-Device</i>
V2G	<i>Vehicle-to-Grid</i>
WIFI	<i>Wireless Fidelity</i>
WHO	<i>World Health Organization</i>
WAVE	<i>Wireless Access in Vehicular Environment</i>
WSMP	<i>WAVE Short Message Protocol</i>

LISTA DE SÍMBOLOS

n	Número de Observadores
$S(i, j)$	Pontuação Atribuída a i e j
l	Similaridade de Luminância
c	Similaridade de Contraste
s	Similaridade Estrutural ou de Correlação
α, β e γ	Constantes Positivas
I	Imagem Original
K	Imagem Compactada
x_n	Rotas A, B e C
W_i	Valor Coerente a x_n
v_i	Conjunto de Veículos
S_{min}	Velocidade Mínima
S_{max}	Velocidade Máxima
$N(b, J)$	Células Disponíveis
μ_x e μ_y	Meios Locais
σ_x e σ_y	Desvios Padrões
σ_{xy}	Covariância Cruzada para as Imagens x e y
α, β e γ	Constantes Positivas
v_i	Conjunto de Veículos
$Dists$	Vetor de Distância
b_j	Associação de Células
M_i	Entradas de QoE, QoS e Distância
m	Número de Elementos Comparados
i	Observador Individual
j	Imagem Classificada Individualmente

$C_{i,j}$	Parâmetros de Maior Prioridade
$N(b_j)$	Classificação das Células Disponíveis
$L(x_i, y_i, t)$	Posição Atual do Veículo
$L(x_i, y_i, t + 1)$	Posição Futura do Veículo
$x_t \in \mathbb{R}^n$	Prever um Estado
H	Fator de Correção
v_k	Fator Ruído
P_k	Covariância de Erro
\hat{x}_k	Coordenadas Previstas
g_{t+1}	Próxima Medição
$K \ n \times m$	Matriz de Ganho

SUMÁRIO

1	INTRODUÇÃO	17
1.1	Contextualização	17
1.2	Motivação e Desafios	18
1.3	Pergunta de Pesquisa	19
1.4	Objetivos	20
1.4.1	Objetivos Gerais	20
1.4.2	Objetivos Específicos	20
1.5	Ineditismo e Contribuições do Trabalho	20
1.6	Organização da Tese	21
2	TECNOLOGIAS CORRELATAS	23
2.1	Redes <i>Ad-hoc</i> Veiculares	23
2.2	<i>Handover</i> em Redes <i>Ad-hoc</i> Veiculares	24
2.3	Avaliações e Métricas de QoE	26
2.3.1	Avaliação Subjetiva de QoE	26
2.3.2	Avaliação Objetiva de QoE	28
2.3.2.1	Índice de Similaridade Estrutural (<i>Structural SIMilarity Index - SSIM</i>)	30
2.3.2.2	Métrica de Qualidade de Vídeo (<i>Video Quality Metric - VQM</i>)	31
2.3.2.3	Outras Métricas de Avaliação de QoE)	33
2.3.3	Avaliação Híbrida de QoE	34
2.4	Processo Hierárquico Analítico - <i>Analytic Hierarchy Process - AHP</i>	35
2.5	Conclusão do Capítulo	38
3	TRABALHOS RELACIONADOS	40
3.1	Trabalhos Correlatos	40
3.2	Conclusão do Capítulo	44
4	ALGORITMO DE <i>HANDOVER</i> SKIP-HOVE	46
4.1	Modelo da Rede e dos Sistemas	46
4.1.1	Etapa de Medição	47
4.1.1.1	Previsão de Mobilidade	48
4.1.1.2	Monitor de QoE	49
4.1.1.3	QoS e Monitor de Sinal	50
4.1.2	Etapa de Decisão	50
4.1.2.1	Cenário de Previsão de Mobilidade	53
4.1.2.1.1	ARIMA	54
4.1.2.1.2	Filtro de Kalman	55
4.1.2.2	Precisão da Previsão de Mobilidade	56
4.2	Conclusão do Capítulo	58
5	AVALIAÇÃO DO ALGORÍTIMO SKIP-HOVE	59

5.1	Descrição e metodologia do cenário	59
5.2	Resultados das Simulações	62
5.3	Conclusão do Capítulo	69
6	CONSIDERAÇÕES FINAIS	71
6.1	Conclusões Gerais	71
6.2	Trabalhos Futuros	72
6.3	Trabalhos Acadêmicos Publicados	72
	REFERÊNCIAS	74
	APÊNDICE A – 1 - LINGUAGEM GNU BASH	79
	APÊNDICE A – 2 - LINGUAGEM C	103

1 INTRODUÇÃO

Neste Capítulo, inicialmente, é realizada a contextualização do tema evidenciando os preceitos conceituais da tese. Na sequência, serão apresentados a motivação e os desafios, bem como os objetivos da tese (gerais e específicos). Essas abordagens relacionam-se à finalidade e às delimitações da pesquisa, tal como a originalidade e contribuição do trabalho, incluindo, a aderência do tema ao Programa de Pós-Graduação em Ciência da Computação da UFPA, e, finalmente, a organização do trabalho.

1.1 Contextualização

As redes de comunicações da próxima geração não serão dependentes apenas de novas tecnologias de acesso, como Múltiplas Entradas e Múltiplas Saídas (*Multiple-input multiple-output* - MIMO)¹ e as Ondas Milimétricas (*Millimeter Wave* - mmWave)², mas também serão utilizadas as infraestruturas de comunicação existentes, como as redes sem fio *Wireless Fidelity* (Wi-Fi) e *Long Term Evolution* (LTE), no intuito primordial de fornecer comunicação onipresente e eficiente (SHAH et al., 2018). Nessa perspectiva, as redes 5G serão compostas por redes de rádio heterogêneas ultradensas que atuarão em conjunto com os sistemas 4G, os quais ocasionalmente aumentam a taxa de dados na borda da rede, conforme ilustrado na Figura 1.

A densificação consiste na implantação maciça de macrocélulas, microcélulas, pequenas células e outras soluções de comunicação, alcançando maior eficiência espectral e a reutilização de taxas do espectro mais altas (DEMARCHOU; PSOMAS; KRIKIDIS, 2018; BEYRANVAND et al., 2017; JI et al., 2018).

Essas células as redes sem fio podem ser usadas para descarregar tráfego de macrocélulas, com o objetivo de permitir a comunicação de todos os tipos de dispositivos em ambientes altamente densos, onipresentes e heterogêneos, possuindo, portanto, um imenso impacto, no que diz respeito aos pontos de vista social e empresarial (MOHAMMADNIA et al., 2019). Por outro lado, a tecnologia sem fio da próxima geração permitirá o uso extensivo de aplicativos de alta demanda, tais como, serviços baseados em vídeo para usuários móveis, a qualquer hora e em qualquer lugar (NIGHTINGALE et al., 2018), incluindo, a distribuição em tempo real de vídeos de publicidade e/ou entretenimento em redes *ad-hoc* veiculares (VANETs).

¹ Fornecer um serviço uniformemente bom aos terminais sem fio em ambientes de alta mobilidade para múltiplos usuários

² Frequências de rádio adequadas para redes 5G. Comparada às frequências abaixo de 5 GHz, usadas anteriormente por dispositivos móveis, a tecnologia de ondas milimétricas permite a transmissão em frequências entre 30 GHz e 300 GHz.

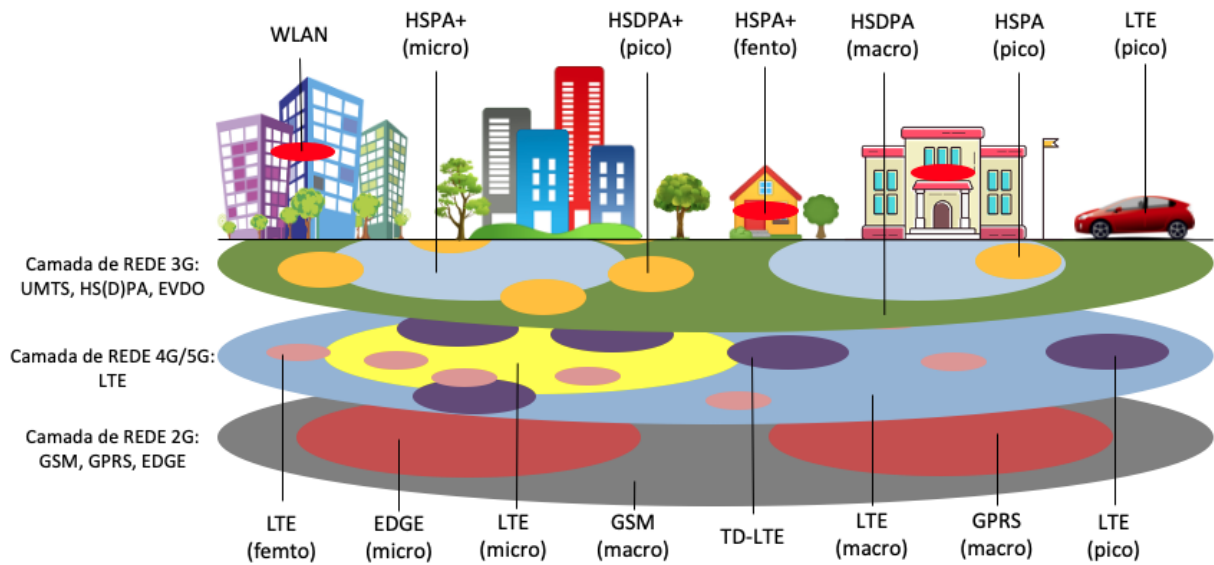


Figura 1 – Exemplo de um cenário VANET Ultradensa.

Fonte: Autor (2020)

1.2 Motivação e Desafios

Uma das questões críticas para o futuro e o sucesso da distribuição de vídeo pelas VANETs, está na capacidade de redes heterogêneas fornecerem mecanismos para prover gerenciamento de recursos eficientes, os quais terão como função aumentar a Qualidade de Experiência (*Quality of Experience - QoE*), otimizando o uso de recursos sem fio e/ou rádio de alta demanda (LUO et al., 2018). No entanto, o aumento do número de células heterogêneas torna o gerenciamento da mobilidade, uma tarefa desafiadora para as VANETs, dado que os veículos, especialmente em ambientes urbanos, geralmente alternam entre diferentes redes heterogêneas. Em outras palavras, quer dizer que os veículos viajam frequentemente de uma área de uma célula para outra, ocasionando muitos *handovers* entre as células disponíveis (NDASHIMYE et al., 2017).

Contudo, o excesso de *handovers* resulta, no chamado efeito de *ping-pong*. Tal efeito significa que um veículo se desconecta de uma célula e depois se conecta novamente na célula anterior (ALJERI; BOUKERCHE, 2019b). Tais problemas relacionados ao *handover* aumentam as perdas de pacotes e/ou quadros de vídeo, levando a uma QoE ruim para aplicações de vídeo em cenários de VANET ultradensa (GUAN et al., 2018). Ignorar *handovers* desnecessários é benéfico para a rede e também para a experiência do usuário (DEMARCHOU; PSOMAS; KRIKIDIS, 2018), já que *handover* bem sucedido promove a conexão de uma unidade móvel de uma célula para outra de forma transparente ao usuário e sem perder a qualidade (MEDEIROS et al., 2018a). Por outro lado, reduzir a frequência de *handovers* acaba renunciando algumas das melhores conexões celulares (ARSHAD et al., 2016).

Dessa maneira, um *handover* eficiente deve manter uma duração do serviço mais longa com a célula, com pelo menos um nível mínimo de QoE, enquanto reduz a sobrecarga de

sinalização e atraso. Por exemplo, uma decisão de entrega baseada na Intensidade do Sinal Recebido (*Received Signal Strength - RSS*)³ desconsidera alguns *handovers*, por mais que signifique que o usuário não esteja recebendo o melhor Sinal Para Interferência Mais Taxa de Ruído (*Signal to Interference Plus Noise Ratio - SINR*)⁴ o tempo todo, acabando por atenuar o efeito *ping-pong* (DEMARCHOU; PSOMAS; KRIKIDIS, 2018). Os esquemas de *handover* baseados em *skip-handover* são frequentemente associados às informações de previsão de mobilidade, com o objetivo de maximizar a duração da conexão sem comprometer o desempenho da rede e/ou aplicativo (ARSHAD et al., 2017; ALJERI; BOUKERCHE, 2019a).

Esse procedimento prioriza as células e dá maior probabilidade para que o usuário permaneça conectado à rede por mais tempo (MOHAMED et al., 2015). Todavia, os esquemas de entrega baseados em *skip-handover* por si só não são suficientes para fornecer vídeos com suporte a QoE, mas, uma decisão de *handover*, baseada também nos parâmetros de QoE e Qualidade de Serviços (*Quality of Service - QoS*) melhora a entrega de vídeo através de VANETs, evitando, assim, *handover* com efeito *ping-pong*. Isso aprimora o uso dos recursos de rede (ARGYRIOU et al., 2017).

A otimização e a utilização dos recursos computacionais presentes em ambientes de VANETs ultradensa, dedicados à distribuição de vídeos, apresentam-se como uma área desafiadora de pesquisa nos últimos anos. Tal desafio se justifica devido à falta de um algoritmo de *handover* eficiente que forneça vídeos com alto QoE. Desta forma, é importante considerar a previsão de mobilidade, a QoS, a QoE, e o *skip-handover* para uma tomada de decisão com maior confiabilidade e menor frequência.

1.3 Pergunta de Pesquisa

Diante de todo o cenário apresentado formulou-se a seguinte "Pergunta de Pesquisa": **Como prover disseminação de vídeos em tempo real em VANETS ultradensas considerando que as entidades que gerenciam e provisionam tais serviços não possuam nenhuma combinação focada, consolidada e associada à predição de mobilidade, QoS, QoE e que resultem em decisões de *handover* entre as células com maior confiabilidade e menor frequência?**

³ O RSS às vezes chamado de RSSI é uma medida da potência presente em um sinal de rádio recebido.

⁴ SINR, é usada para fornecer limites teóricos superiores à capacidade do canal (ou à taxa de *handover* de informações) em sistemas de comunicação sem fio.

1.4 Objetivos

1.4.1 Objetivos Gerais

Esta tese de doutorado tem como objetivo propor um algoritmo de *handover* baseado em múltiplos critérios e decisões de *skip-handover* para distribuição de vídeo em tempo real em cenários de VANETs ultradensas, chamado de Skip-HoVe. O algoritmo Skip-HoVe garante conexão sem interrupções em VANETs ultradensas, com a finalidade de entregar vídeos com alta QoE, levando em consideração a previsão de mobilidade, a QoS e a QoE, resultando em decisões de *handover* com maior confiabilidade e menor frequência. O Skip-HoVe considera um método multicritério de apoio às decisões, chamado de Processo de Análise Hierárquica (*Analytic Hierarchy Process - AHP*), no sentido de atribuir diferentes graus de importância para cada critério. Ademais, o algoritmo também evita proativamente o efeito *Ping-Pong* por meio de decisões baseadas em *skip-handover*, ou seja, ignorando *handovers* quando a QoE e a QoS forem aceitáveis e estáveis.

1.4.2 Objetivos Específicos

Para alcançar os objetivos delineados anteriormente, os seguintes objetivos específicos foram considerados, planejados e realizados:

- Analisar fatores que impactam na QoE da distribuição de vídeos em tempo-real em ambientes de VANET ultradensas;
- Analisar a distribuição de vídeo em tempo real baseado em *handover* em VANETs no contexto de redes ultradensas;
- Elaborar e validar um algoritmo de *handover* multicritério para distribuição de vídeos em VANET em redes ultradensas, denominado de Skip-HoVe;
- Testar duas técnicas de previsão de mobilidade com o algoritmo Skip-HoVe, a saber: i) Média Móvel Integrada Autorregressiva (*AutoRegressive Integrated Moving Average - ARIMA*); e Filtro Kalman (*Kalman Filter - KF*);
- Minimizar o efeito *ping-pong* em ambientes VANETs baseados em *skip-handover*;
- Avaliar o desempenho do algoritmo Skip-HoVe para distribuição de vídeo em tempo real em uma VANET ultradensa.

1.5 Ineditismo e Contribuições do Trabalho

Com o propósito de validar a originalidade e ineditismo do trabalho proposto, no que diz respeito a averiguar trabalhos relacionados ou similares ao propósito, foi realizada uma busca na

extensa literatura de algoritmos de *handover*. Porém, foram encontrados apenas trabalhos que conseguem resolver o problema parcialmente, tal como será apresentado no Capítulo 3.

Em relação às principais contribuições deste trabalho, tem-se o resumo abaixo:

- Um algoritmo de *handover* denominado de Skip-HoVe, o qual tem como objetivo maximizar o tempo de conexão com uma célula de serviço;
- Uma tomada de decisão com base em vários critérios para decisões de *handover*, para disseminação de vídeo com suporte à QoE em um cenário de VANETs ultradensas.
- Uma técnica de ignorar as decisões de *handovers* quando a QoE e a QoS forem aceitáveis e estáveis, de forma que evite o efeito *ping-pong*.
- Resultados da simulação, com a finalidade de apresentar o desempenho do Skip-HoVe na distribuição de vídeos com suporte a QoE em cenários VANETs ultradensas, comparando aos dos algoritmos de *handover* já existentes. Por exemplo, os resultados de Pontuação de Opinião Média (*Mean Opinion Score* - MOS) apresentaram uma melhora de 30% nas avaliações subjetivas, enquanto a taxa de *handover* considerada como *ping-pong* foi mantida abaixo de 2%.

1.6 Organização da Tese

A estrutura desta tese é organizada por seis capítulos, em uma sequência lógica na seguinte forma:

- Capítulo 2: É apresentada a fundamentação teórica com o propósito de fornecer subsídios para o entendimento desta pesquisa, abordando previamente conceitos sobre tecnologias relacionada às VANETs. A QoE é abordada juntamente com suas avaliações subjetivas, objetivas e híbridas. O modelo AHP é discutido com exposição de exemplos baseados em tomadas de decisões complexas.
- Capítulo 3: É apresentada uma síntese do levantamento bibliográfico realizado sobre os principais algoritmos de *handover* existentes para cenários em VANETs, assim como as vantagens e desvantagens de cada uma dessas algoritmos.
- Capítulo 4: É apresentado o algoritmo de *handover* Skip-HoVe que considera diversos critérios para distribuição de vídeo em cenários de VANETs ultradensa. Também são apresentadas duas técnicas de previsão de mobilidade com o algoritmo de *handover*, a saber: ARIMA e KF.
- Capítulo 5: São descritos os resultados da simulação do algoritmo Skip-HoVe e as melhorias alcançadas em relação aos algoritmos de *handover* existentes.

- Capítulo 6: São descritos as considerações finais deste trabalho, demonstrando a eficiência do novo algoritmo baseado em vários critérios para distribuição de vídeo em VANETs ultradensas, bem como os trabalhos futuros e os principais artigos publicados relacionados a este trabalho.

2 TECNOLOGIAS CORRELATAS

Esse capítulo descrever as principais características e desafios das tecnologias relacionadas à área central dessa tese, ou seja, à disseminação de vídeos em VANETs ultradensas. A QoE também é discutida, bem como suas métricas de avaliação objetivas, subjetivas e híbridas e a importância de uma avaliação em cenários que envolvem distribuição de conteúdo multimídia. O modelo para tomada de decisões em AHP é explicado em detalhes, com uma demonstração completa em uma decisão genérica. O objetivo da revisão compreende a verificação do estado da arte e a identificação de oportunidades de pesquisa sobre a temática. Logo, serão apresentadas as mais atuais e principais tecnologias utilizadas e a pesquisa efetuada, entre outros aspectos relacionados.

2.1 Redes *Ad-hoc* Veiculares

As Redes *Ad-hoc* Veiculares (*Vehicular Ad-hoc Networks* - VANETs) possuem uma grande variedade de aplicações para segurança humana, além de auxiliarem os motoristas a conduzirem seus veículos em estradas, tanto em regiões urbanas, quanto rurais. Um dos objetivos das VANETs é prover comunicação para proporcionar uma viagem segura, evitando situações perigosas causadas por acidentes na estrada. Além desse fato, as VANETs fornecem comunicação entre os veículos em movimento e a infraestrutura, a fim de transferir diferentes tipos de dados, tal como emergenciais, *Infotainment/SoftNews*, entretenimento, monitoramento inteligente, broadcast de anúncios e propagandas, etc., como mostrado na Figura 2

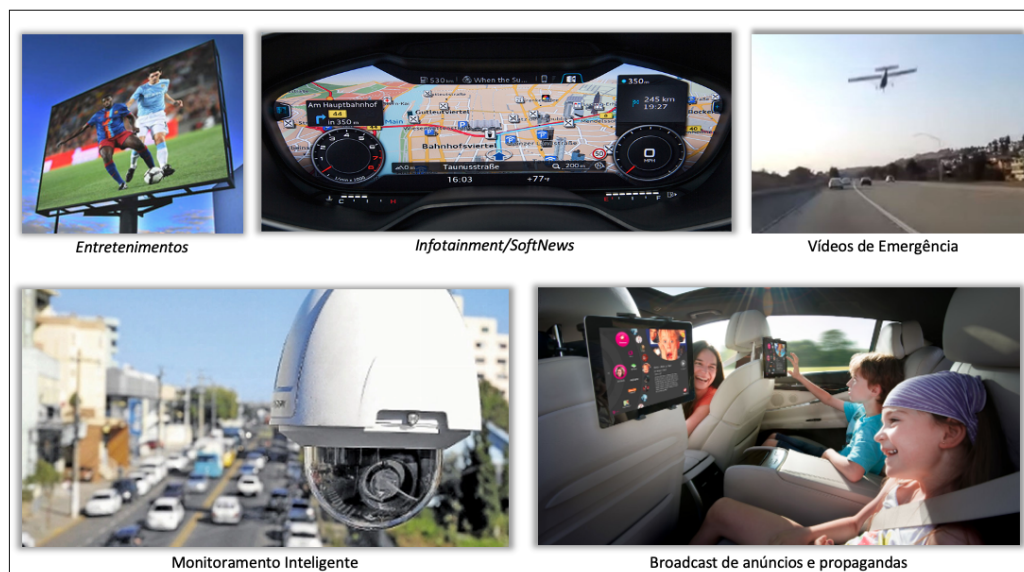


Figura 2 – Aplicações de vídeos em tempo real e sob demanda.

Fonte: Autor (2020)

Nas VANETs, os veículos são equipados com sensores especializados, conhecidos como Unidade de Bordo (*On-BoardUnit* - OBU), nos quais coletam automaticamente informações

em tempo real dos veículos em movimento. As VANETs consistem em comunicação Veículo para Veículo (*Vehicle-to-Vehicle - V2V*), Veículo para Borda de Estrada (*Vehicle-to-Roadside - V2R*), Veículo para Infraestrutura (*Vehicle-to-Infrastructure - V2I*), Veículo para Pedrastes (*Vehicle-to-Pedestrian - V2P*), Veículo para Dispositivos (*Vehicle-to-Device - V2D*), Veículo para Malha (*Vehicle-to-Grid - V2G*) e híbrida com recursos exclusivos, conforme observado na Figura 3 A topologia de rede muda rapidamente para os cenários VANETs, assim como o tamanho de rede, a troca de dados sensíveis ao atrasos e outros fatores que a tornam diferente de outras redes ad hoc, como as Rede *Ad-hoc* para Celular (*Mobile ad hoc Network - MANETs*).

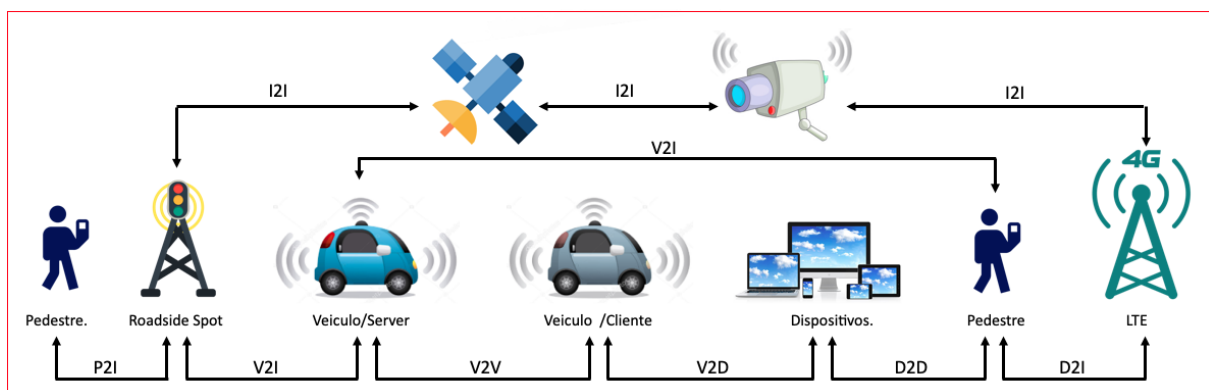


Figura 3 – Topologia Dinâmica das Redes V2V, V2I, V2P e outras

Fonte: Autor (2020)

2.2 Handover em Redes Ad-hoc Veiculares

O *handover* é uma técnica de gerenciamento de mobilidade utilizada em redes de comunicação sem fio, para tratar, de forma transparente para o usuário, a transição de uma Unidade Móvel (*Mobile Unity - MU*) para outra área de cobertura ou célula em caso de enfraquecimento do sinal. A Figura 4 mostra um conjunto de veículos locomovendo-se pelas células t_1 , t_2 e t_3 de uma estação base (5G BS) para outra (LTE BS). A intensidade média do sinal da 5G BS diminui a medida que os veículos se afastam da respectiva BS. Da mesma forma, a intensidade média do sinal da LTE BS aumenta a medida que o veículo se aproxima. O *handover* ocorre durante a transição entre as BSs.

Handover em cenários VANETs significam mudança no Ponto de Fixação (*Point of Attachment - PoA*) de um nó móvel (OBU) ou fixo (RSU). Portanto, o *handover* visa manter as conexões ativas quando o OBU muda seu ponto de conexão. O *handover* é a técnica utilizada no intuito de melhorar a mobilidade em redes ad hoc, considerando que, os veículos se comunicam com as RSUs diretamente, bem como, por meio de outros OBUs, respectivamente. De acordo com (THAKUR; GANPATI, 2019), um processo de *handover* consiste em três etapas principais, sendo:

- **Medição:** Uma estação móvel descobre várias redes sem fio com base em anúncios de

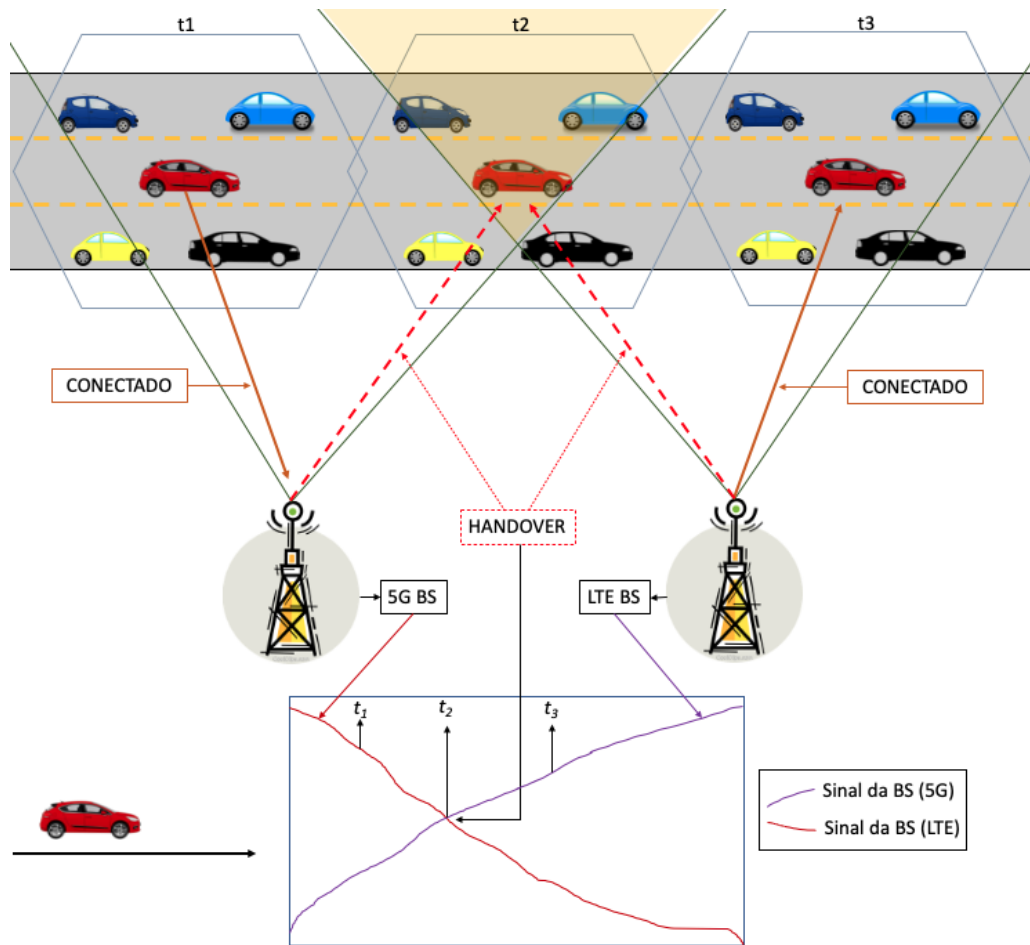


Figura 4 – Tendências no Processo de Handover
Fonte: Autor (2020)

serviços transmitidos por essas redes. A unidade móvel procura mensagens nos canais atribuídos e cria uma lista de pontos de acesso priorizados pela intensidade do sinal recebido e outros parâmetros. O método de medição é categorizado por dois padrões: (i) A varredura ativa, onde a estação não apenas ouvirá as mensagens provenientes dos pontos de acesso, mas também enviará mensagens para eles. (ii) Varredura passiva, onde a estação apenas ouve as mensagens dos demais dispositivos conectados.

- **Decisão:** Após coletar os dados de medição, a estação decidirá, com base em diversos critérios, quando e para quem a entregados dados deve ser realizada.
- **Execução:** Com base nos parâmetros coletados na etapa de Decisão, ocorre o handover real, onde a rede atual transfere para a próxima rede as informações de roteamento necessárias e outras informações contextuais sobre a estação. Por exemplo, um veículo se movimenta de uma célula para outra adjacente e a rede faz o *handover* da conexão para esta nova célula.

O *handover* é um elemento essencial de comunicação que deve ser considerado no

desenvolvimento de sistemas para VANETS. O uso de algoritmos eficientes de *handover* é um modo rentável de aumentar a capacidade e o QoS de sistemas celulares.

2.3 Avaliações e Métricas de QoE

A QoE tem recebido muita atenção nos últimos anos, tornando-se, então, uma questão importante para a qualidade de diversos serviços, tal como o vídeo em tempo real. Para tal objetivo, uma vasta pesquisa na literatura foi realizada para entender e medir a QoE designada a uma variedade de serviços multimídia. Logo, a finalidade é melhorar e gerenciar a qualidade dos serviços multimídia e, ao mesmo tempo, garantir as operações de rede eficientes e econômicas (NAGIN et al., 2019).

A avaliação da QoE do usuário final tornou-se um dos principais empecilhos para atender às altas exigências tecnológicas, tais como, a satisfação do usuário em qualquer circunstância, o que significa receber seja qual for o serviço, a qualquer hora, em qualquer lugar, independente do dispositivo, por qualquer mídia e tecnologia de rede.

Estudos recentes e desafios emergentes relacionados ao gerenciamento de QoE para serviços multimídia em VANETs fornecem uma visão geral do campo de abordagens de avaliação de QoE, incluindo, portanto, dois aspectos observados na Figura 5, que são a avaliação subjetiva e a avaliação objetiva.

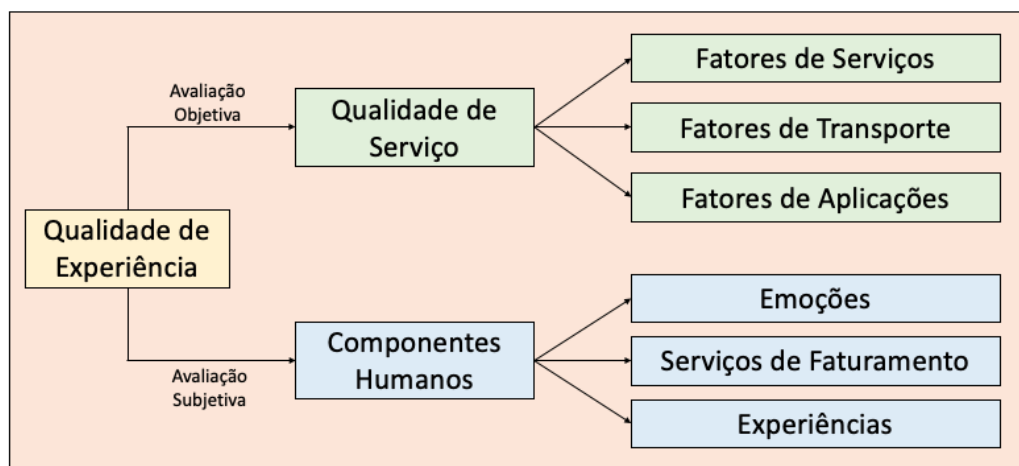


Figura 5 – Diagrama de Qualidade de Experiência.

Fonte: Autor (2020)

2.3.1 Avaliação Subjetiva de QoE

As avaliações subjetivas da QoE são baseadas em experimentos psicoacústicos/visuais que representam a maneira fundamental e mais confiável e complexa para a QoE dos avaliadores (ROSÁRIO et al., 2016; CERQUEIRA et al., 2014). Esses métodos foram pesquisados e investigados por muitos anos e permitiram que os pesquisadores obtivessem uma compreensão

mais profunda das dimensões subjetivas da QoE. Mais comumente, os resultados de qualquer experimento subjetivo são classificações de qualidade dos usuários, obtidas durante sua avaliação ou posterior a ela, que são então calculadas pelo MOS. Essa abordagem foi especificada na Recomendação P.800.1 da ITU-T (UNION, 2006) e uma classificação de qualidade média de um grupo de usuários foi expressada por meio de escalas padronizadas. O MOS foi estendido a outros procedimentos de avaliação subjetiva recomendados pela ITU e classificados pelo tipo de aplicação e mídia (ITU, 2010), como notado na Tabela 1.

Tabela 1 – Tabela Descritiva do MOS

MOS		
MOS	Qualidade	Descrição
9-10	EXCELENTE	Nenhum defeito pode ser percebido pelo observador ao avaliar o vídeo.
7-8	BOM	Defeito no vídeo é perceptível, porém não causa desconforto ao avaliador.
5-6	REGULAR	Defeito é percebido no vídeo e causa desconforto ao avaliador.
3-4	RUIM	Grande degradação no vídeo, porém o avaliador consegue visualizar alguma informação no vídeo.
1-2	PÉSSIMO	Vídeo torna-se ininteligível e o avaliador fica impossibilitado de extrair informação.

Uma pontuação média é calculada para obter o MOS para uma imagem específica usando a Equação 2.1, onde n indica o número de observadores e $S(i, j)$ é a pontuação atribuída pelo observador i à imagem j .

$$MOS(j) = \frac{1}{n} \sum_{i=1}^n S(i, j) \quad (2.1)$$

Embora o MOS seja usado como uma medida subjetiva de desempenho, ele baseia-se na conversão das quantidades objetivas em pontuações subjetivas. É uma abordagem utilizada para a avaliação de parâmetros de qualidade pelos usuários e por meio de escalas padronizadas.

Os testes subjetivos geralmente são baseados em experiências controladas da vida real com participantes humanos, pagos ou voluntários que avaliam diretamente sua experiência em um aplicativo ou serviço. Esses usuários podem estar envolvidos no experimento de maneira passiva (apenas visualizando e/ou ouvindo) ou de maneira ativa (participando de uma conversa) e julgam a qualidade em relação à apresentação de algum estímulo. Por exemplo, os participantes podem ser chamados para avaliar a qualidade de um vídeo em tempo real através de um serviço telefônico. Esses testes precisam ser cuidadosamente projetados com antecedência e o grupo de usuários precisa ser selecionado adequadamente com base nas diretrizes e recomendações dos órgãos de padronização (ITU-T; RECOMMEND, 1996).

Várias técnicas podem ser usadas para avaliação subjetiva. Por exemplo, os usuários podem avaliar a qualidade usando uma escala de classificação absoluta ou podem comparar vídeos/imagens/áudios de forma sequencial indicando qual possui a melhor experiência na avaliação. Os resultados baseados em opiniões de usuários quantificam principalmente a eficácia, a eficiência e a satisfação geral do conteúdo avaliado. Esses tipos de testes subjetivos são considerados os mais confiáveis, pois incorporam diversos aspectos conscientes e inconscientes da avaliação da qualidade humana, principalmente se forem cuidadosamente projetados e se os usuários forem imparciais e objetivos.

Uma maneira de superar esse problema é realizar uma avaliação de QoE de "serviço real", na qual os usuários avaliam sua experiência durante a excussão e após o término do serviço. Esse exemplo é o paradigma "OneClick", que pode ser usado em tempo real para monitoramento, *feedback* e controle da QoE. Essa estrutura exige apenas que o sujeito clique na tecla dedicada sempre que se sentir insatisfeito com a qualidade do aplicativo em uso (CHEN; TU; XIAO, 2009). Com base em experimentos psicológicos, a abordagem mais benéfica a ser utilizada é por meio da avaliação subjetiva da qualidade, mas também a mais complexa e dispendiosa para avaliar a QoE dos usuários. A abordagem é iniciada a partir da percepção do usuário ao avaliar a operação (HUSIĆ; BARAKOVIĆ; OSMANOVIĆ, 2019).

2.3.2 Avaliação Objetiva de QoE

Diversos modelos de avaliação objetiva de QoE já foram desenvolvidos, sendo que cada um deles apresenta suas próprias circunstâncias e restrições. Considerando que não existe um modelo universal definido, os modelos propostos podem ser classificados em diferentes critérios (RECOMMENDATION, 2013). As métricas de avaliação objetiva de QoE são modelos matemáticos que buscam resultados próximos dos das métricas subjetivas. As métricas objetivas são divididas em três tipos, apresentados a seguir:

- **Sem Referência (*No Reference - NR*)** Estas métricas dependem de medições de parâmetros da sequência de saída do vídeo para avaliar a qualidade objetiva e não utilizam nenhuma informação do sinal original. Sua complexidade computacional é a mais baixa de todas as métricas, obtendo alta praticabilidade e instantaneidade, a sua precisão também é menor. Ademais, a métrica de NR estão condicionadas à características específicas, as quais não são utilizadas em sentido geral, como nos serviços *on-line*, onde o conteúdo original não está disponível (CHEN; JIN; YANG, 2016). A avaliação objetiva de QoE através das métricas NR é mostrada na Figura 6.
- **Referência Reduzida *Reduced Reference - RR*** -Essa métrica compara os parâmetros parciais selecionados a partir da entrada original e da sequência de saída do sistema para, assim, obter a avaliação da qualidade. Os parâmetros podem estar na camada de aplicação (taxa de bits e taxa de quadros) ou na camada de rede (perda de pacotes). As métricas RR

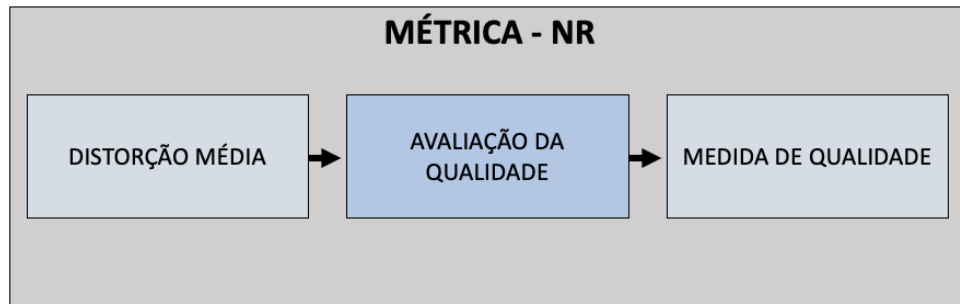


Figura 6 – Avaliação objetiva de QoE através das métricas NR

Fonte: Autor (2020)

precisam apenas extrair parâmetros específicos da transmissão da sequência de entrada para o usuário, através de canais auxiliares sem erros, sem solicitar toda a sequência. Essa métrica é adequada a *handover* em tempo real pela rede e possuem baixa largura de banda e de transmissão. Além disso, a métrica RR são mais condicionadas à uma situação a qual a sequência de entrada original fica difícil de ser transportada e armazenada, (YILMAZ, 2018). A avaliação objetiva de QoE através das métricas RR é mostrada na Figura 7.

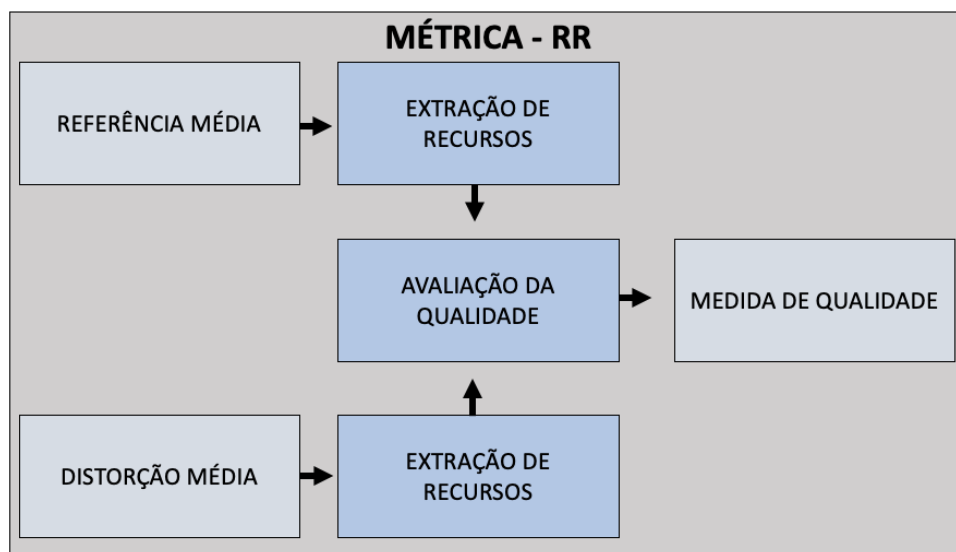


Figura 7 – Avaliação objetiva de QoE através das métricas RR.

Fonte: Autor (2020)

- Referência Completa (Full Reference - FR)** - Essa métrica compara a sequência de saída do sistema (sequência de teste) com a sequência de entrada original (sequência de referência). O resultado da avaliação tem alta precisão, no entanto, exige maior processamento computacional. A métrica FR mais utilizadas são: (i) Erro Médio Quadrático (*Mean Squared Error - MSE*), Relação Sinal Ruído de Pico (*Peak Signal to Noise Ratio - PSNR*), Índice de Similaridade Estrutural (*Structural Similarity Index - SSIM*) e a Métrica de Qualidade de Vídeo (*Video Quality Metric - VQM*) (DÍAZ et al., 2020), (MURUDKAR; GITLIN, 2019) e (SUNG et al., 2019). A avaliação objetiva de QoE através das métricas FR é mostrada na Figura 8.

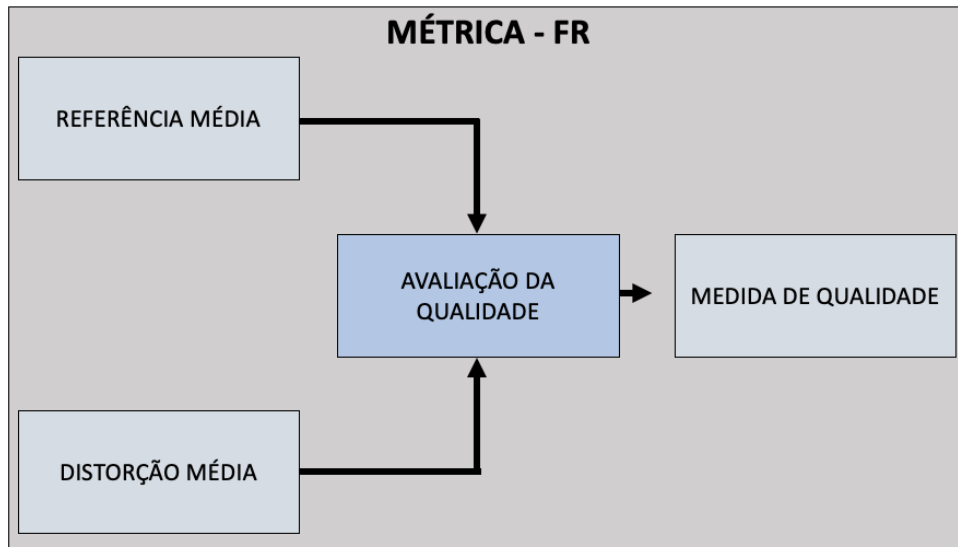


Figura 8 – Avaliação objetiva de QoE através das métricas FR.

Fonte: Autor (2020)

Assim, em resumo, a métrica FR podem ser usadas para todas as referências. Os sinais de vídeo enviados e recebidos são comparadas entre si. No entanto, devido à sobrecarga causada pelo sistema, essa métrica não pode ser usada em cenários reais. A métrica RR é usado na transmissão de dados adicionais, mesmo que reduzidos. As métricas NR surgiram para superar uma sobrecarga no sistema.

As métricas de avaliação objetivas também são classificadas segundo a abordagem utilizada para estimar a qualidade do sinal. As Subseções 2.3.2.1 e 2.3.2.2 destacam duas métricas de avaliação objetiva de vídeo, sendo as métricas SSIM e VQM, respectivamente (COSTA et al., 2011). Já a Subseção 2.3.2.3 apresentam outras métricas também aplicadas em avaliação objetiva da qualidade.

2.3.2.1 Índice de Similaridade Estrutural (*Structural SIMilarity Index - SSIM*)

O SSIM é um modelo baseado na percepção. A degradação da imagem é considerada como mudança de percepção nas informações estruturais. O mascaramento de luminância acontece quando parte da distorção de uma imagem é menos visível em suas bordas. Já o mascaramento por contraste acontece quando as distorções da imagem também são menos visíveis em sua textura. O termo informação estrutural enfatiza *pixels* fortemente interdependentes ou *pixels* fechados espacialmente. Esses *pixels* fortemente interdependentes se referem a algumas informações mais importantes sobre os objetos visuais no domínio da imagem.

O SSIM estima a qualidade percebida de imagens e vídeo, e mede a similaridade entre duas imagens: a original e a recuperada. Essa similaridade estrutural em múltipla escala é calculada considerando mudanças de luminância, de contraste e de estrutura. O SSIM é avaliado base no cálculo de três aspectos principais denominados luminância, contraste e estrutural. (SARA; AKTER; UDDIN, 2019). A luminância compara o brilho entre duas imagens. O contraste

diferencia os intervalos entre as regiões mais claras e mais escuras de duas imagens. Finalmente, a estrutura compara o padrão de luminância local entre duas imagens para encontrar a semelhança e a dissimilaridade das imagens. Estes aspectos são independentes. Por exemplo, uma mudança de luminosidade e contraste não afetam a estrutura da imagem. Este índice é uma combinação da multiplicação desses três aspectos que pode ser expresso através da Equação 2.2. As letras l , c e s representam a luminância, o contraste e a estrutura respectivamente. As constantes positivas são representadas pelas letras gregas α , β e γ .

$$SSIM(x, y) = [l(x, y)]^\alpha \cdot [c(x, y)]^\beta \cdot [s(x, y)]^\gamma \quad (2.2)$$

Sendo assim, as similaridades de luminância, similaridade de contraste e similaridade estrutural de uma imagem podem ser expressas separadamente de acordo com as Equações 2.3, 2.4 e 2.5, onde μ_x e μ_y são os meios locais, σ_x e σ_y são os desvios padrões e σ_{xy} é a covariância cruzada para as imagens x e y sequencialmente.

$$Luminancia \equiv l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \quad (2.3)$$

$$Contraste \equiv c(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \quad (2.4)$$

$$Estrutural \equiv e(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \quad (2.5)$$

Se $\alpha = \beta = \gamma = 1$, o índice é simplificado através das Equações 2.3 e 2.4.

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_x\sigma_y + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (2.6)$$

A Equação 2.6 e expressada na escala em decibel dB como $10\log_{10}[SSIM(x, y)]$. O SSIM produz um valor decimal entre 0 e 1, onde 0 significa correlação zero com a imagem original e 1 significa exatamente a mesma imagem.

2.3.2.2 Métrica de Qualidade de Vídeo (*Video Quality Metric* - VQM)

A métrica VQM, VQM fornece abordagens padronizadas e não padronizadas para medir a qualidade percebida do vídeo, considerando aspectos estruturais e temporais. A implementação da VQM inclui três etapas: extração de características baseadas em percepção, computação de parâmetros de qualidade de vídeo e a combinação desses parâmetros para finalizar o modelo geral.

A VQM é uma métrica FR que analisa um vídeo original e outro distorcido como entrada e retorna um valor de distorção D entre 0 (sem distorção) e 5 (distorção máxima).

A métrica VQM utiliza a Transformada Coseno Discreta (*Discrete Cosine Transform - DST*)¹ para corresponder à percepção humana a qual é muito utilizada em processamento digital de imagens e compressão de dados (LARRANAGA C et al., 2016). O algoritmo para calcular a métrica VQM é descrito a seguir:

1. Receber os vídeos original e processado, ambos com conteúdos distintos.
2. Realizar a transformação de cores YUV criadas a partir de fontes RGB (vermelho, verde e azul), onde os valores ponderados de R , G e B são somados para produzir Y' , uma medida de brilho ou luminância geral.
3. Realizar a transformação de DCT, na qual as imagens recebidas são separadas em diferentes componentes de frequência espacial.
4. Converter cada coeficiente de DCT em Contraste Local (*Local Contrast - LC*) através da Equação 2.7.

$$LC(i, j) = DCT(i, j) \frac{\frac{DC}{\text{mean}(DCT(i, j))}}{DC}^{\delta} \quad (2.7)$$

$$LC(i, j) = DCT(i, j) \left(\frac{DC}{1024} \right)^{0,65} \frac{1}{DC} \quad (2.8)$$

O componente de cada bloco é denominado Corrente Direta (*Direct Current - DC*). Para uma imagem de 8 bits o valor médio de DCT é 1024, conforme demonstrado na Equação 2.8. O melhor parâmetro para a montagem de dados de psicofísica é $\delta = 0,65$. Após essa etapa, a maioria dos valores fica entre $[-1, 1]$ (CHO et al., 2007). De acordo com as Equação 2.7 e 2.8, a conotação $LC(i, j)$ é convertida para assim ser multiplicada pela entrada do coeficiente de DCT correspondente na matriz de Função de Sensibilidade ao Contraste Espacial (*Spatial Contrast Sensitivity Function - SCSF*).

5. Calcular a subtração sobre cada imagem processada.
6. Incorporar a máscara de contraste com uma operação máxima simples e pesá-la com a distorção média ponderada, resultando em um agrupamento ponderado de distorções média (D_m) e máxima (D_{max}), de acordo com as Equações 2.9 e 2.10, respectivamente, onde I é a imagem original e o K a imagem compactada. O valor de 1000 corresponde a padronização do rádio.

¹ DCT expressa uma sequência finita de pontos de dados em termos de uma soma de funções de cosseno que oscilam em diferentes frequências. O DCT, proposto inicialmente por Nasir Ahmed em 1972, é uma técnica de transformação amplamente usada no processamento de sinais e compactação de dados.

7. Finalmente, calcular a métrica VQM através da Equação 2.11. O valor 0,005 corresponde ao parâmetro de máximo de distorção, o qual é um valor constante com base em estudos psicovisuais.

$$D_{\mu} = 1000\text{mean}(|I - K|) \quad (2.9)$$

$$D_{max} = 1000\text{mean}(|I - K|) \quad (2.10)$$

$$VQM = D_{max} + 0,0005D_{max} \quad (2.11)$$

O O algoritmo do cálculo da VQM possui a maior complexidade comparada às dos algoritmos das demais métricas de avaliação objetiva, mas também apresenta a melhor correspondência com as métricas de avaliação subjetiva, nas quais os resultados obtiveram um coeficiente de correlação de 0.95 de ganho em alguns testes realizados (KOTEVSKI; MITREVSKI, 2010).

2.3.2.3 Outras Métricas de Avaliação de QoE)

Existem diversas métricas de avaliação objetiva de qualidade de vídeos que não são aplicadas neste trabalho, porém vale destacar algumas delas.

- Relação Sinal-Ruído de Pico - *Peak Signal to Noise Ratio* (PSNR) é uma das métricas mais utilizadas pelos pesquisadores. A PSNR avalia a diferença entre o sinal original e o distorcido, calculando o erro quadrático médio entre os dois sinais e, assim, identifica a relação entre a potência máxima possível de um sinal e a potência do ruído de interrupção.
- Escala Multipla-SSIM - (*MultiScale-SSIM* - MS-SSIM) é uma extensão do SSIM de única escala, originalmente direcionada para imagens estáticas e posteriormente estendida para vídeo. Essa extensão considera a densidade de amostragem da imagem, a distância entre o observador e a imagem, e a capacidade perceptiva do Sistema Visual Humano (*Human Visual System* - HVS) do observador. O MS-SSIM resume a influência de cada escala com diferentes pesos para considerar a sua importância relativa.
- Métrica de Qualidade de Imagem em Movimento - (*Moving Picture Quality Metric* - MPQM) é a métrica mais utilizada para avaliar a qualidade de imagem em movimento, simulando o modelo HVS no espaço-temporal com uma abordagem de banco de filtros. Ela também, considera duas características da percepção humana: sensibilidade ao contraste e efeito de mascaramento, para explicar a detecção visual. A métrica MPQM avalia uma sequência de vídeo melhor que a métrica SSIM e não apenas as imagens de quadro único, dado que considera a influência da qualidade de vídeo afetada por fatores relacionados à transmissão na rede.

- Fidelidade à Informação Visual - (*Visual Information Fidelity* - VIF) tem a finalidade de avaliar a qualidade da imagem, comparando dois tipos de informação: o conhecimento da imagem de referência (quando a imagem original passa diretamente pela HVS) e a informação da imagem distorcida (quando a imagem original passa pela primeira vez através do canal de distorção, em seguida, para o HVS).
- Relação Sinal-Ruído Visual - (*Visual Signal-to-Noise Ratio* - VSNR) consiste em avaliar a qualidade apresentada em imagens estáticas, além de possuir um desempenho prospectivo para avaliação de qualidade de vídeo. Esta métrica determina distorções quase limítrofes e supralimiais da visão humana com o objetivo de minimizar o problema supraliminar na HVS. Ela também aplica conceitos visuais de mascaramento e soma para confirmar distorções detectáveis.
- Avaliação de Integridade de Vídeo Baseada em Movimento - (*Motion-based Video Integrity Evaluation* - MOVIE) detecta a distorção de vídeo no domínio espaço-temporal através da computação das trajetórias, apresentadas em vídeo. Inclui ainda dois componentes, sendo estes os índices MOVIE espacial e temporal, os quais medem as distorções espaciais. O valor final da métrica MOVIE para sequência de vídeo é calculado como um produto de ambos os índices.

2.3.3 Avaliação Híbrida de QoE

Conforme descrito nas subseções 2.3.1 e 2.3.2, as avaliações subjetiva e objetiva de qualidade têm várias limitações, sendo que a híbrida é a mais recente em termos de avaliação de QoE. Muitas pesquisas foram feitas para abordar essas limitações conforme relacionam os autores (GHAREEB; VIHO, 2010), os quais determinam em suas pesquisa que a forma mais apropriada para avaliar a QoE no contexto do *streaming* de vídeo é através de várias sobreposições de Redes de Distribuição de Vídeo - (*Video Distribution Networks* - VDN). Com esse objetivo, os autores compararam três tipos diferentes de avaliação de QoE, sendo a subjetiva, objetiva e híbrida que pode superar as limitações das duas primeiras.

Já os autores (MEDEIROS et al., 2018b) introduziram mecanismos de baixa complexidade para avaliar a QoE de forma híbrida que, através de um determinado veículo X conseguem estimar a QoE de um fluxo de vídeo. Os autores (GREENGRASS; EVANS; BEGEN, 2009) destacam o impacto que diferentes durações de perda de pacotes têm na QoE dos serviços de *streaming* de vídeo baseados em IP. Eles descrevem as deficiências visuais resultantes dessas perdas de pacotes, tanto para os serviços padrões de alta definição quanto para os codificados em MPEG-2. A avaliação híbrida de QoE que os citados autores pesquisaram mede o nível de qualidade em tempo real através das informações dos cabeçalhos de pacotes de *codecs* de IP e de vídeo. De uma forma mais detalhada, a avaliação híbrida de QoE desenvolvida será discutida na Subseção 4.1.1.2, onde serão abordadas todas as suas técnicas e funcionalidades.

2.4 Processo Hierárquico Analítico - *Analytic Hierarchy Process - AHP*)

O Processo Hierárquico Analítico (*Analytic Hierarchy Process - AHP*) é um dos principais modelos matemáticos atualmente disponíveis para apoiar a teoria da decisão. Embora o AHP possa ser utilizado por critérios delimitados a decisões simples, ele pode ter melhor resultado para tomada de decisões quando apresentado a problemas mais complexos, especialmente aqueles de decisões labiríntica, as quais exigem uma percepção mais apurada e cuja resolução terá repercussão a longo prazo, visto que o AHP se baseia em decisões complexas, incorporando, portanto, evidências quantitativas e resultados qualitativos para promover a construção de um consenso estruturado. O AHP pode suplementar e/ou melhorar a transparência das contribuições de resultados para o desenvolvimento de diretrizes ou critérios (WANG; WANG; LIU, 2017).

Assim, o uso do AHP apresenta vantagens quando elementos da decisão são difíceis de quantificar ou comparar, ou quando a comunicação entre os mesmos apresenta obstáculos por especialidades, terminologias ou perspectivas diferentes. O uso do AHP para este trabalho pode ser exemplificado da seguinte forma:

- Identificação das alternativas e atributos significantes sobre os resultados prévios;
- Os responsáveis pela decisão indicam a significância relativa entre os atributos. Por exemplo, se as alternativas são referentes às decisões de um trajeto de um carro em uma via, o condutor pode preferir o trajeto mais longo do que o percurso mais próximo, porém engarrafado;
- Similarmente, para cada atributo, bem como para cada par de alternativas, os responsáveis pela decisão especificam suas preferências (por exemplo, se a distância do percurso da alternativa A for menor que a do percurso da alternativa B);
- As comparações entre os atributos e as alternativas são registradas em matrizes na forma de frações entre 1/9 e 9. Cada matriz é avaliada pelo seu autovalor com o objetivo de verificar a coerência dos julgamentos. Este procedimento gera uma "razão de coerência", que será igual a 1 se todos os julgamentos forem coerentes entre si. Caso um tomador de decisão, afirmar que prefere X a Y, Y a Z, contudo, ele preferir Z a X, então a razão de coerência será maior do que 1,2 e os julgamentos precisarão ser revisados. Este passo é uma das principais razões para que muitos usuários do AHP acreditem que, se trata de um processo com bom embasamento teórico;
- Valores globais de preferência são calculados para cada alternativa.

O AHP é muito usado em simulações de sistemas que transformam a análise qualitativa em análise quantitativa. Ele pode efetivamente evitar as dificuldades e determinar o peso de

cada fator quantitativo e qualitativo. O AHP pode também manter a consistência do processo de julgamento e é amplamente utilizado para determinar o peso que expressa a importância relativa de um conjunto de alternativas com base em vários critérios. Neste estudo, o AHP foi selecionado para determinar o peso de cada fator, a fim de obter um índice de classificação abrangente das características dos dados coletados em VANETs para a tomada de decisões sobre manutenção do nível da rede.

Dessa forma, o AHP pode ser aplicado em várias decisões com problemas que envolvem um número finito de alternativas. Formalmente, nesse contexto, em um processo de decisão existe um conjunto finito de alternativas, $X = \{x_1, \dots, x_n\}$, a partir do qual o tomador de decisão é geralmente motivado a escolher a melhor opção. A seguir, será apresentada uma situação hipotética para exemplificar o processo de tomada de decisão baseada no AHP.

Um determinado usuário de VANET precisa decidir qual o trajeto percorrer durante uma viagem. Resumidamente, o objetivo do usuário é a sua melhor satisfação ao seu destino final. Este exemplo é demonstrado na Equação 2.12.

$$X = \underbrace{\{RotaA\}}_{x_1}, \underbrace{\{RotaB\}}_{x_2}, \underbrace{\{RotaC\}}_{x_3} \quad (2.12)$$

Regulamente, nos processos de decisão, o tomador de decisão atribui uma pontuação a cada alternativa e, em seguida, escolhe aquela com o valor máximo. Ou seja, dado um conjunto de alternativas, $X = \{x_1, \dots, x_n\}$, o tomador de decisão deve fornecer ao vetor um peso de acordo com a Equação 2.13, onde w_i é um valor que estima coerentemente a pontuação da alternativa x_i . Ou seja, quanto maior w_i , melhor será o valor em i .

$$w = (w_1, \dots, w_n)^T \quad (2.13)$$

O autor (BRUNELLI, 2014) afirma que estes fatores são coerentes com a teoria dos valores. A regra é que a alternativa x_i é preferida à alternativa x_j se e somente se $w_i > w_j$. Os vetores de ponderação nada mais são do que classificações e seus componentes w_i são prioridades ou pesos chamados das alternativas x_i . Por exemplo $w = (0.4, 0.2, 0.3, 0.1)^T$, onde as entradas dos pesos implicam em $x_1 \succ x_3 \succ x_2 \succ x_4$ onde o $x_i \succ x_j$ significa que a alternativa x_i é preferida a x_j . Os possíveis vínculos são expressos como $x_i \sim x_j$.

Considere o exemplo da escolha da melhor rota. Se o vetor $w = (0.3, 0.5, 0.2)^T$ estivesse associado ao conjunto de alternativas $X = \{RotaA, RotaB, RotaC\}$ então tem-se a seguinte classificação $RotaB \succ RotaA \succ RotaC$ devido ser $w_2 > w_1 > w_3$. Tomar decisões dessa maneira parece fácil, mas se torna uma tarefa difícil quando a complexidade aumenta devido à quantidade de alternativas e de critérios.

O tomador de decisões pode ter dificuldades quando solicitado a enviar uma classificação na forma de um vetor numérico para um grande número de alternativas, comprometendo assim a

decisão final. Essa questão se origina nos limites cognitivos e da impossibilidade de comparar efetivamente várias alternativas simultaneamente. Uma maneira eficaz de superar esse problema é usar comparações pareadas. Assim, a estratégia é decompor o problema original em pares, e escolher o mais apropriado.

As comparações aos pares são coletadas na matriz de comparação entre as partes $A = (a_{ij})_{n \times n}$, conforme descrito na Equação 2.14.

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \quad (2.14)$$

A estrutura da matriz é composta por $a_{ij} > 0$ que expressa o grau de preferência de x_i a x_j . Mais precisamente, de acordo com a teoria de tomada de decisão baseada no AHP (SAATY, 2008) seu maior benefício é uma abstração da estrutura de um sistema para estudar as interações funcionais de seus componentes e seus impactos no sistema total, onde cada entrada deve aproximar a relação entre dois pesos, conforme expressa a Equação 2.15.

$$a_{ij} \approx \frac{w_i}{w_j} \forall i, j \quad (2.15)$$

Isto significa que, se as entradas representam proporções entre os pesos, a matriz A quando contabilizada com a Equação 2.15, resulta na Equação 2.16, pois esta é uma condição de multiplicação da reciprocidade mantida através da consequência intuitiva e quase direta da condição que $a_{ij} = 1/a_{ji} \forall i, j$, isto é $i, j = 1, \dots, n$. Note que é uma condição imposta quando constroem-se essas matrizes.

$$A = (w_i/w_j)_{n,n} = \begin{pmatrix} w_1/w_1 & w_1/w_2 & \cdots & w_1/w_n \\ w_2/w_1 & w_2/w_2 & \cdots & w_2/w_n \\ \vdots & \vdots & \ddots & \vdots \\ w_n/w_1 & w_n/w_2 & \cdots & w_n/w_n \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix} = n \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix} \quad (2.16)$$

A matriz A pode ser simplificada conforme a Equação 2.17.

$$A = \begin{pmatrix} 1 & a_{12} & \cdots & a_{1n} \\ \frac{1}{a_{12}} & 1 & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{a_{1n}} & \frac{1}{a_{2n}} & \cdots & 1 \end{pmatrix} \quad (2.17)$$

A estrutura simplificada das matrizes de comparação por pares nesta forma segue a suposição de que, por exemplo, se x_1 é 2 vezes melhor que x_2 , então x_2 é 1/2 tão bom quanto x_1 .

A Equação 2.18 expõe a exemplificação com a hipotética situação de classificar uma matriz de comparação pareada para o conjunto de rotas $X = \{RotaA, RotaB, RotaC\}$.

$$A = \begin{pmatrix} 1 & 3 & 6 \\ 1/3 & 1 & 2 \\ 1/6 & 1/2 & 1 \end{pmatrix} \quad (2.18)$$

A partir dessa matriz, em particular da entrada a_{12} , pode-se descobrir que x_1 (Rota A) é considerado três vezes melhor que x_2 (Rota B). Ou seja, $a_{12} = 3$ sugere que $w_1 = 3w_2$.

$$w = \begin{pmatrix} 6/9 \\ 2/9 \\ 1/9 \end{pmatrix} \quad (2.19)$$

No exemplo, pode-se verificar que se a condição $a_{ij} = w_i/w_j, \forall i, j$ é satisfeita pelo vetor a seguir com seus componentes somados até um, então a RotaA (x_1) é classificada como a melhor. Resumindo, sempre que o número de alternativas for muito grande, compará-las aos pares é uma maneira eficaz de se obter uma classificação. A classificação das alternativas contidas na matriz w agora é mais robusta do que teria sido se tivesse sido estimada diretamente, sem usar a matriz A .

2.5 Conclusão do Capítulo

Como visto neste Capítulo, o principal objetivo das VANETs é assegurar uma viagem segura a motoristas e passageiros, buscando evitar a incidência de acidentes nas estradas. Além disso, também possibilita a comunicação entre veículos em movimento, através do repasse de dados entre eles. Um dos pontos relevantes abordados foi na Subseção 2.2, que corresponde a uma técnica utilizada em redes sem fio para permitir uma comunicação mais precisa por meio da transição transparente de uma área de cobertura para uma célula, em caso de enfraquecimento do sinal, , melhorando, assim mobilidade em redes *ad-hoc*.

Acerca das métricas de QoE existem as métricas de avaliações subjetiva e objetiva, cada uma com suas próprias circunstâncias e restrições. Há três tipos de métricas de avaliação subjetiva: as NRs, que dependem de medições de parâmetros da sequência de saída; as RRs, que comparam parâmetros parciais selecionados a partir da entrada original e da sequência de saída do sistema; e, por fim, as FRs, que comparam tanto a referência quanto o resultado da sequência de entrada original com a sequência de saída do sistema. Ademais, as principais métricas de avaliação de vídeo são a SSIM, a qual avalia uma sequência de imagens estáticas considerando o fenômeno de mascaramento visual ao medir a semelhança entre as imagens original e recuperada, e a métrica VQM que estabelece abordagens padronizadas e não padronizadas para medir a

qualidade não percebida do vídeo, considerando seus aspectos temporais e estruturais. A métrica VQM possui um algoritmo mais elaborado que os das demais métricas.

Este Capítulo também abordou a importância da aplicação do AHP, nesta tese, baseado tomada de decisões complexas, dado que possui evidências quantitativas e resultados qualitativos com a funcionalidade primordial de estabelecer critérios extremamente importantes. Nesta trabalho, sua função principal é manter a consistência do processo de julgamento ao determinar a importância relativa de um conjunto de alternativas baseadas em diversos critérios diferentes. Sendo assim, o AHP pode ser utilizado para a tomada de decisões que possuem várias alternativas e vários caminhos que podem ser percorridos, diante da possibilidade de cada escolha. Logo, ele auxilia o usuário VANET a decidir, por exemplo, o melhor trajeto para sua viagem, considerando as infinitas possibilidades.

3 TRABALHOS RELACIONADOS

Este Capítulo apresenta uma revisão do estado da arte com o objetivo de identificar oportunidades de pesquisa relacionada a algoritmos de *handover* para distribuição de vídeos em VANETs ultradensas. Por fim, é apresentado um quadro comparativo entre os algoritmos pesquisados.

3.1 Trabalhos Correlatos

Os autores (PACHECO et al., 2019) enfatizam que as VANETs que atuam na rede 5G são uma solução promissora e fornecem os requisitos para uma grande quantidade de aplicativos, como a disseminação de vídeos em tempo real. No entanto, um alto número de células implantadas nos cenários 5G causa uma grande quantidade de desconexões e *handover*, podendo comprometer a QoE do usuário final. Dessa forma, os autores apresentam um algoritmo preditivo de entrega de QoE com previsão de mobilidade para redes veiculares chamado HoVe. Nesse estudo, fica claro que o gerenciamento eficiente da mobilidade em VANET na rede 5G é um dos principais desafios enfrentados pela próxima geração de comunicação móvel, onde a tendência de densificação da rede transcende as limitações das gerações anteriores, mas aumenta a complexidade necessária no plano de gerenciamento. As técnicas de previsão apresentadas constituem uma ferramenta essencial para manter os usuários em veículos conectados e com QoE suficiente em todos os seus percursos. Já os algoritmos de gerenciamento de mobilidade de última geração não garantem a entrega adequada de conteúdo multimídia. Todavia, os algoritmos apresentados pelos autores possuem parâmetros simples de previsão de mobilidade na decisão de *handover*, que consideram a QoE e a qualidade do sinal na tomada de decisões multicritério. Os resultados da simulação dos autores mostram a eficácia da técnica, com uma melhoria de 18% em relação aos algoritmos de última geração.

Os autores (ROSÁRIO et al., 2013) relatam que a convergência de aplicativos multimídia em tempo real, a crescente cobertura de redes sem fio heterogêneas e a crescente popularidade dos dispositivos móveis estão levando a uma era de serviços multimídia móveis centrados no usuário. Nesse cenário, as comunicações heterogêneas coexistem e garantem que o usuário final esteja sempre melhor conectado. As rigorosas demandas de rede dos sistemas multimídia sem fio, além das estratégias de controle orientadas para a qualidade, são necessárias para garantir a melhor experiência do usuário ao longo do tempo. Dessa forma, os autores enfatizam que se faz necessário o suporte à QoE. Tal estudo propõe uma arquitetura de *handover* com QoE para redes sem fio heterogêneas convergentes, denominada QoEHand, onde o algoritmo estende o Entrega Independente de Mídia - (*Media Independent Handover* - MIH) / IEEE 802.21 com reconhecimento de QoE, mobilidade e adaptação de vídeo, integrando um conjunto de módulos de tomada de decisão. Em suma, os autores introduziram seu algoritmo para aprimorar a percepção humana e otimizar o uso de recursos em sistemas sem fio convergentes competitivos

com baixa complexidade e processamento, coordenando os esquemas de estimador, mapeamento e adaptações de qualidade. Devido à sua abordagem ser no formato modular, o que torna o experimento muito interessante, o QoEHand pode ser ajustado para operar com diferentes tecnologias, como LTE e classes de serviço em redes sem fio. Nas simulações dos autores, foram realizados testes a fim de demonstrar o impacto e os benefícios do QoEHand em um sistema multiusuário baseado nos padrões IEEE 802.11e e IEEE802.16e. Seus resultados mostraram que o algoritmo fornece um melhor nível de qualidade para aplicativos de vídeo em tempo real em comparação com o MIH puro, revelando um excelente MOS durante os experimentos.

(GONG et al., 2018), destacam que a utilização de Rede Ultradensas Heterogênea - (*Heterogeneous Ultra-Dense Network - H-UDN*) leva a uma maior taxa de falha de *handover* entre camadas. Assim, a otimização da estratégia tradicional de *handover* é necessária e urgente. Os autores relatam que, ao aplicar combinações de resultados analíticos estocásticos de longo prazo com parâmetros instantâneos e tomando esses resultados como orientação, eles adotaram como fatores de decisão a força do sinal recebido, o tempo de permanência na cobertura de pequenas células e a carga de tráfego da BS. Para integrar de maneira abrangente esses três fatores de decisão e escolher a BS de *handover* do alvo, os autores aplicaram a análise baseada no Processo Hierárquico Analítico Difuso - (*Fuzzy Analytical Hierarchical Process - FAHP*) e a correlação cinza em sua estratégia. Como resultado, o FMHD traz melhorias no desempenho do *handover* entre camadas, o que pode ser observado na redução da probabilidade de falha no *handover* e do efeito *ping-pong* em redes heterogêneas e ultradensa, definindo um *Time-To-Trigger* (TTT) durante a execução do *handover*. Ou seja, nas simulações foi avaliado o processo de *handover* entre camadas, aplicando o FMHD e o algoritmo tradicional separadamente, alterando o TTT e a velocidade do usuário. Assim, foi observada a redução da probabilidade de falhas no *handover* e probabilidade de efeito *ping-pong*, além de aumentar a taxa de dados. Além disso, o FMHD proposto pelos autores pode ser aplicado em diversos tipos de implantação de *handover*.

Embora seja destacada a importância de uma decisão de *handover* de vários parâmetros, o uso do TTT pode ter efeitos indesejados, como falhas de *link* e atrasos no processo de *handover* (LEE et al., 2010). Cabe registrar que nesta mesma citação, os autores usam o TTT a fim de mitigar o efeito de *handover* sobre o efeito *ping-pong*, que é uma troca desnecessária da célula de serviço por um curto período de tempo. Por outro lado, a taxa de Falha no Link de Rádio - (*Radio Link Failure - RLF*) pode aumentar devido ao atraso na execução do *handover* durante o TTT. Portanto, valores TTT eficientes de acordo com os tipos de células vizinhas e as velocidades do UE precisam ser investigados para alcançar simultaneamente a baixa taxa de *ping-pong* e a baixa taxa de RLF.

Os autores (SILVA; BECVAR; FRANCES, 2018) propuseram um método de *handover* adaptativo usando o TTT e lógica Fuzzy com base na velocidade do veículo do usuário. O algoritmo de entrega coleta parâmetros de mobilidade para prever a localização do usuário para,

assim um conteúdo, mostrando que o descarregamento de macrocélulas para células pequenas pode ser essencial em um ambiente heterogêneo. Um dos principais benefícios do esquema proposto é a redução das taxas de *ping-pong* em cenários densos. O algoritmo proposto pelos autores, apresenta desempenho de *handover* ao dos métodos tradicionais e das soluções mais avançadas atualmente. A solução proposta quase elimina o efeito de *ping-pong* e, além disso, também reduz a taxa de falha no *handover*, preservando assim os benefícios das Estações Base de Células Pequenas - (*Small Cellbase Stations* - SCellBs), enquanto os indicadores de desempenho de entrega total são notavelmente aprimorados. Isso indica a adequação do algoritmo proposto para futuras redes móveis com uma densidade muito alta nas SCellBs. No entanto, ele não se destina ao tráfego de multimídia e, portanto, não considera QoE para tomada de decisão.

Os autores (LIU et al., 2019) introduziram um TTT adaptável para minimizar o impacto de *handover* frequente em redes ultradensas. O trabalho mencionado, aplica uma decisão do Fuzzy TOPSIS para escolher o melhor candidato a *handover*, a fim de atingir níveis adequados de QoS. Cabe destacar, que a pesquisa em si apresenta um algoritmo *handover* que considera mais de um atributo como critério para acionar o processo de *handover* e que seleciona a opção mais apropriada. Esse algoritmo de *handover* proposto combina os pontos fortes do TOPSIS e da lógica fuzzy. Além disso, para aprimorá-lo ainda mais, os autores, utilizaram do esquema de subtração de *cluster* para gerar função de associação difusa inteligente usando dados históricos. Os resultados da simulação indicaram que o algoritmo de *handover* proposto reduz significativamente o número de futuros *handover* e a taxa de efeito *ping-pong* na rede, mantendo a QoS em um nível relativamente alto, comparado com o algoritmo baseado em RSS e a abordagem de Algoritmos de Decisão de Vários Atributos - (*Multiple Attributes Decision Algorithms* - MADM) convencional. A técnica de subtração de *cluster* proposta pode definir de forma inteligente a função ótima de associação difusa a partir de dados históricos para diferentes critérios. Todavia, ela não aplica métricas preditivas ou qualquer monitoramento de QoE, o que melhoraria significativamente a qualidade das decisões em um cenário da VANET.

Os autores (ARSHAD et al., 2017) demonstram que o *handover* introduz uma sobrecarga na rede, o que às vezes a torna redundante e que ignorar alguns *handovers* pode ser benéfico para a rede, mantendo uma QoS aceitável. Este estudo busca analisar o efeito do atraso de *handover* através de um modelo matemático baseado na geometria estocástica e validado através de simulações. Os resultados apresentados refletem os impactos negativos de *handover* na taxa de usuários em redes celulares densas e enfatizam o potencial dos esquemas de *handover* propostos para mitigar esse impacto. Analiticamente, os resultados apresentados mostram ganhos de até 77%, quando comparados aos do esquema de *handover* convencional, que sempre mantém a melhor associação de RSS. No entanto, esse trabalho oferece apenas um pequeno suporte à transmissão de vídeo e pode não ser adequado aos requisitos estritos envolvidos.

Já os autores (DEMARCHOU; PSOMAS; KRIKIDIS, 2018) propuseram duas técnicas a fim de ignorar dinamicamente altas taxas de *handovers* na rede. Em primeiro lugar, os autores

utilizam uma técnica *skip-ping* da NC que considera a associação de uma BS única e que tem por base três critérios: (a) a área da célula, (b) a distância da trajetória dentro da célula e (c) a distância da BS da borda da célula. Seus resultados mostram que, para velocidades moderadas do usuário, a técnica Não Cooperativo - (*non-cooperative* - NC) oferece a maior média de rendimento. Em segundo lugar, a técnica de salto baseado em Cooperativo - (*cooperative* - CO) refere-se a um usuário que desfruta de benefícios de cooperação com BS, a menos que um *handover* seja ignorado. A decisão de *handover* é tomada com base em três células consecutivas do padrão de trajetória do usuário. Todavia, este trabalho considera a previsão de trajetória na decisão dos *handover*, assumindo um modelo simples baseado apenas em posição e velocidade.

Em outra pesquisa os autores (XU et al., 2019) propuseram um algoritmo de *handover* de camada cruzada orientada pelo atraso, a fim de maximizar o desempenho de aplicações de baixa latência em redes ultradensas. O trabalho, baseou-se em uma expressão analítica para atingir a capacidade adequada dos usuários durante a execução de *handover*, propondo um esquema de alocação de recursos nas células alvo e assim, reduzir a probabilidade de bloqueio. Entretanto, este estudo apresentou também um modelo de otimização no desempenho de camadas intermediárias em redes heterogêneas a partir da perspectiva das características de atraso. A rede é descrita pela geometria estocástica como uma rede heterogênea de duas camadas, na qual a capacidade efetiva é considerada e que pode refletir o atraso na taxa. Para os diferentes tipos de requisitos de QoS dos usuários, eles são divididos em duas categorias: usuário sensível ao atraso e usuário sensível à taxa. Com base na teoria da otimização convexa, é realizada a otimização conjunta da capacidade efetiva e da probabilidade de bloqueio. As simulações dos autores provam que, com o aumento de usuários sensíveis ao atraso, o fator de probabilidade de *handover* diminuir e os recursos de largura de banda reservados aos usuários aumentam. O algoritmo de otimização reduz a probabilidade de bloqueio para os usuários sensíveis ao atraso, enquanto mantém a capacidade efetiva relativamente constante. Os resultados das simulações, a modelagem matemática e a teoria que os autores abordam podem ser aplicados à estratégia de *handover* e a seleção da estratégia de reserva de recursos em redes heterogêneas, porém os autores não empregam esquemas preditivos ou informações de mobilidade na decisão, o que pode melhorar a qualidade da decisão e impactar positivamente na QoE do usuário.

Nos estudos (MEDEIROS et al., 2018b) e (MEDEIROS et al., 2018a), os autores confirmaram a importância da tomada de decisão de *handover* multicritério para equilibrar métricas de diferentes camadas (Medições de Rádio, QoS e QoE) utilizando AHP de acordo com os níveis de importância predefinidos atribuídos a cada um. Contudo, o algoritmo apresenta altas taxas de *handover*, o que é prejudicial à QoE em cenários densos.

Na pesquisa realizada pelos autores (SARGENTO et al., 2016), foi proposto um gerenciador de conexões para as VANETs com tecnologias heterogêneas baseada em AHP chamado Gerenciador de Conexões VANET - (*VANET Connection Manager* - VCM), que combina informações de várias fontes (velocidade do veículo, GPS, cabeçalho, RSSI e tecnologias disponíveis,

como DSRC/WAVE, IEEE 802.11 e Celular 4G), decidindo, então, a melhor conexão disponível em todos os momentos, tentando também minimizar o número de *handovers*. O AHP é otimizado através da interação com Algoritmos Genéricos - (*Genetic Algorithm* - GA). Essa abordagem inclui a previsão de mobilidade durante o tempo de conectividade esperado, não incluindo os requisitos de QoE.

Já os autores (ZHANG et al., 2018) apresentam em seus estudos uma classificação de aplicativos sensíveis e insensíveis com base na experiência do usuário. Nessa perspectiva, quando previsto, durante o tempo ocioso de uma rede de alto desempenho, uma decisão de *handover* muda para outra rede mais eficiente energeticamente.

Finalmente, os pesquisadores (CHEN et al., 2018) apresentam uma estimativa de QoE para correlacioná-la com a QoS, com o objetivo de melhorar a satisfação do usuário sem se concentrar apenas na probabilidade de bloqueio de chamadas, mas também, na probabilidade de falha no processo de *handover*. No entanto, o compartilhamento de vídeo requer métricas mais subjetivas para descrever a QoE, como o MOS, que pode ser imitado por algoritmos de aprendizado de máquina e integrado a decisões automatizadas.

3.2 Conclusão do Capítulo

Com base na análise do estado da arte, conclui-se que a distribuição de vídeo em cenários de VANETs ultradensas requer um algoritmo eficiente, baseado em *skip-handover* para manter um número mínimo de interrupções e evitar a ocorrência do efeito *ping-pong*.

Tabela 2 – Quadro Comparativo de Trabalhos Relacionados

RELAÇÃO DOS TRABALHOS RELACIONADOS AVALIADOS					
#	Referências	Técnica Utilizada	QoE	Handover	Predição de Mobilidade
01	(PACHECO et al., 2019)	AHP	✓	✓	✓
02	(ROSÁRIO et al., 2013)	MIH	✓	✓	✓
03	(GONG et al., 2018)	FAHP	✗	✓	✗
04	(LEE et al., 2010)	TTT Adaptável	✗	✓	✗
05	(SILVA et al., 2018)	TTT Adaptável	✗	✗	✗
06	(LIU et al., 2019)	Logica Fuzzy	✗	✗	✗
07	(ARSHAD et al., 2017)	Handover Skipping	✗	✓	✗
08	(DEMARCHOU 2018)	Handover Skipping	✗	✓	✓
09	(XU et al., 2019)	Orientado a Atrasos	✓	✓	✗
10	(MEDEIROS et al., 2018b)	AHP	✓	✗	✗
11	(SARGENTO et al., 2016)	AHP	✗	✓	✓
12	(ZHANG et al., 2018)	Q Learning	✓	✗	✗
13	(CHEN et al., 2018)	Q Learning	✓	✗	✗

Fonte: Autor (2020)

A Tabela 2 resume as principais características dos algoritmos de *handover* analisados

em termos de reconhecimento de QoE e de previsão de mobilidade.

Após o estudo sobre o estado da arte, nota-se que, no cenário atual, uma técnica eficiente de previsão de mobilidade é necessária para melhorar as decisões de *handover*. Além disso, é vital considerar um esquema de decisão com vários critérios, para, assim, equilibrar as métricas heterogêneas que impactam de forma direta e indireta na experiência do usuário quanto este faz uso de serviços de vídeo.

4 ALGORITMO DE *HANDOVER* SKIP-HOVE

Este Capítulo apresenta um algoritmo de *handover* que possui diversos critérios para distribuição de vídeo em cenários de VANETs ultradensas, chamado Skip-HoVe (COSTA et al., 2020). O objetivo é mitigar os efeitos adversos de *handover* frequentes, mantendo um nível aceitável de QoE dos vídeos distribuídos. Uma condição proativa é empregada para impedir que haja o efeito *ping-pong* durante a decisão de *skip-handover*. Além disso, esta decisão evita o *handover* quando a QoE e a QoS são aceitáveis e estáveis na conexão corrente, considerando sempre a preferência pelas células que maximizem o tempo conectado.

4.1 Modelo da Rede e dos Sistemas

Esta tese considera um cenário composto por um conjunto de n veículos $V = \{v_1, v_2, \dots, v_n\}$ com uma identidade individual $i \in [1, n]$. Cada veículo v_i tem um transceptor de rádio para permitir a comunicação V2V e V2I, sendo que esta última exige infraestruturas para suportar as redes veiculares, pois, nesse tipo de comunicação, os veículos só trocam informações com as infraestruturas às margens das vias.

Para a comunicação V2I, uma rede celular ultradensa é considerada como uma rede da camada K , na qual cada camada modela a célula de acesso específica, como macrocélula, microcélula e picocélula. Nesse sentido, é considerado um conjunto de células $B = \{b_1, b_2, \dots, b_m\}$ com uma identidade individual $j \in [1, m]$, implantadas em locais fixos e conhecidos (x_j, y_j) . As células entre as camadas podem diferir-se em termos de densidade espacial e potência de transmissão P_j . Também é assumido uma rede principal com fibras de alta capacidade conectadas nesta infraestrutura para evitar congestionamentos nos enlaces de retorno. Denota-se $N(b_j) \subset B$ como um subconjunto de células dentro do intervalo de rádio (R_{max}) de um determinado veículo v_i . Em relação ao conteúdo do vídeo, cada vídeo compactado é composto por três tipos de quadros: I, P e B (GREENGRASS; EVANS; BEGEN, 2009; ROSÁRIO et al., 2016). Esses quadros são organizados em sequências, chamadas de Grupo de Figuras - (*Group of Pictures* - GoP), que apresentam todas as informações necessárias para decodificar um determinado vídeo em um período (ROSÁRIO et al., 2013). Ademais, denota-se um determinado Fluxo de Vídeo (*Video Flow* - VF) por $(VF_i = g_1, g_2, \dots, g_k)$ como um conjunto de GoP, k e g , onde cada quadro em um determinado GoP g é dividido em um ou mais pacotes de vídeo p , dependendo do tamanho de cada quadro (QUADROS et al., 2017). Cada pacote p contém, além da carga útil de dados, outros parâmetros do codificador, tais como: sinalizador de tipo de quadro, ID, comprimento, tempo (data e hora) e segmentação de pacote (GREENGRASS; EVANS; BEGEN, 2009).

Para obter essas informações, um monitor de pacote no lado do cliente extrai as informações sobre o tipo de quadro e sobre a dependência intraquadro para cada pacote p (NOIRIE et al., 2009), dado que cada VF começa com um cabeçalho de sequência, seguido por um cabeçalho

GoP e depois por um ou mais quadros codificados. Outrossim, cada veículo v_i pode medir a qualidade do sinal recebido através de um parâmetro de rádio de cada célula disponível $N(b_j)$, usando a medida chamada de Qualidade do Sinal de Referência Recebido - (*Reference Signal Received Quality* - RSRQ).

Cada veículo v_i é ciente de sua localização $L(x_i, y_i, t)$ em um determinado tempo t , usando um sistema de posicionamento, tal como o GPS. Além disso, cada veículo v_i viaja seguindo uma determinada trajetória tra_j com uma velocidade s_i variando entre um limite de velocidade mínimo (Ex, s_{min}) e outro máximo (Ex, s_{max}). Singularmente, o veículo v_i desloca-se sobre diferentes áreas devido ao seu rápido movimento e, portanto, frequentemente possui um conjunto diferente de células disponíveis $N(b_j)$.

A Entidade de Gerenciamento de Mobilidade - (*Mobility Management Entity* - MME) é a principal entidade controladora no Núcleo do Pacote Evoluído - (*Evolved Packet Core* - EPC) para a rede de acesso LTE. Neste caso, a entidade gerenciadora de *handover* executa todas as seguintes etapas de entrega:

- Etapa de Medição;
- Etapa de Decisão;
- Etapa de Execução (se necessário).

A entidade gerenciadora deve ter uma conexão com cada célula b_j , sendo uma interface de sinalização S1. Um veículo v_i logicamente se comunica com o gerente de entrega através de seu b_j atual para relatar as medições que podem ser solicitadas pelo gerenciamento de mobilidade, se necessário. Cada veículo v_i realiza medições, enquanto o MME executa a decisão e o *handover*. Na etapa de medição, a entidade gerenciadora obtém informações de v_i e $N(b_j)$. Posteriormente, na etapa de decisão, a entidade considera as informações coletadas na etapa anterior para selecionar o b_j ao qual um determinado veículo v_i deve conectar-se. Finalmente, na etapa de execução, a entidade gerenciadora da entrega é responsável por alterar a conexão de um determinado veículo v_i de uma célula de atendimento para uma célula de destino, escolhida pela entidade. Mais detalhes sobre cada etapa são apresentados nas próximas subseções.

4.1.1 Etapa de Medição

Na etapa de medição, o algoritmo Skip-HoVe coleta informações do veículo v_i , e de suas células disponíveis $N(b_j)$. Especificamente, essas informações do veículo v_i são sua QoE estimada, sua localização atual, sua QoS e seus parâmetros de rádio. O Skip-HoVe também coleta informações de QoS e QoE das células de veiculação e candidatas $N(b_j)$ para entender seu desempenho e tomar a melhor decisão. A entidade gerenciadora atribui valores máximos de QoS e QoE quando a célula está ociosa para que esta seja a preferida e, assim, fornecer balanceamento

de carga. A Figura 9 representa algumas das funções desempenhadas pelo gerenciamento de *handover*. Tal gerenciamento controla a infraestrutura da rede (macrocélula, microcélula e picocélula) e acessa seus componentes internos, tais como o Gerenciamento de Mobilidade e os Algoritmos de *Handover*.

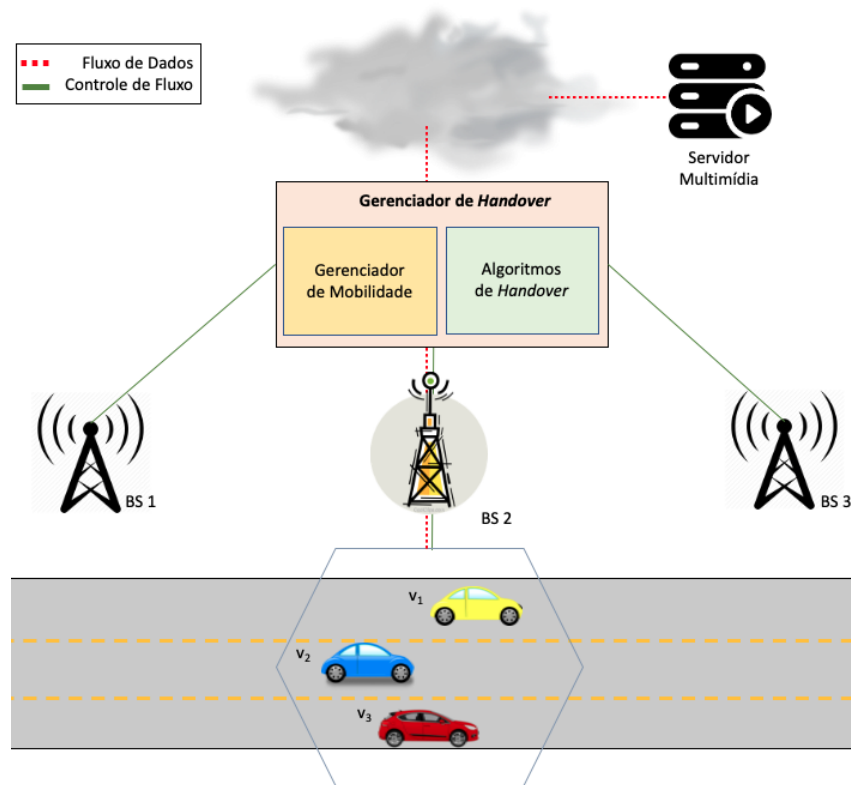


Figura 9 – Etapas de Gerenciamento de *Handover*

Fonte: Autor (2020)

4.1.1.1 Previsão de Mobilidade

A mobilidade de um veículo é aproximadamente linear, aumentando, assim, a sua precisão de previsão (BUI et al., 2017). Nesse sentido, um algoritmo de previsão de mobilidade, tal como ARIMA ou KF (BUI et al., 2017), estima a posição $L(x_i, y_i, t + 1)$ de um determinado veículo v_i em um tempo futuro $t + 1$ com base na sua velocidade e na sua localização através de equações da Cinemática. deste, fazendo o uso de equações cinemáticas. Com base em tal previsão, é possível evitar conexões com uma célula que pode não estar mais disponível no futuro. Logo, torna-se torna útil tratar a mobilidade como uma série temporal, na qual cada medida constitui uma entrada para o preditor ajustar o modelo de previsão.

A granularidade da previsão, em um contexto espacial e temporal, pode ser definida por uma frequência de medição. Por exemplo, o algoritmo Skip-HoVe executa uma nova previsão a cada nova medição. Outrossim, também é avaliada a granularidade da previsão de mobilidade com variação de 0,1 a 2 segundos em etapas de 0,2 segundos. A respeito dessa avaliação, adota-se uma granularidade de 1 segundo, dado a simplicidade do módulo de previsão, o qual não causou

sobrecarga significativa. No entanto, para outros cenários, esse valor pode ser parametrizado para ajustar os recursos de mobilidade e computação adequadamente.

A posição prevista é baseada no ponto do espaço onde o veículo provavelmente estará na próxima medição. Segundo a localização prevista do veículo, o algoritmo Skip-HoVe calcula a distância $d_{i,j}$ entre a posição futura do veículo $L(x_i, y_i, t + 1)$ e suas células disponíveis $N(b_j)$. Uma grande distância significa que uma célula está indo além do veículo e isso deve ser evitado. No entanto, valores maiores correspondem a uma pontuação mais alta para esses b_j durante uma decisão de *handover*. Portanto, o vetor de distância $Dists$ é invertido, conforme mostrado na Equação 4.1.

$$Dists = \left[\frac{1}{d_{i,0}} \quad \frac{1}{d_{i,1}} \quad \frac{1}{d_{i,2}} \quad \cdots \quad \frac{1}{d_{i,3}} \right] \quad (4.1)$$

Vários valores no vetor $Dists$ podem estar próximos de zero devido às grandes distâncias. Logo, o vetor $Dists$ deve ser normalizado dividindo cada elemento pelo valor absoluto do vetor $Dists$, calculado com base na Equação 4.2.

$$|Dists| = \sqrt{(d_{i,0})^2 + (d_{i,1})^2 + \cdots + (d_{i,2})^2}. \quad (4.2)$$

Esses valores podem ser inseridos no algoritmo ao calcular o ponto d na célula individual b_j com base na Equação 4.3.

$$d = \frac{d_{i,j}}{|Dists|}, \forall d_{i,j} \in Dists. \quad (4.3)$$

4.1.1.2 Monitor de QoE

O algoritmo Skip-HoVe apresenta um monitor de QoE híbrido de baixa complexidade executando, através de um determinado veículo v_i com o objetivo de estimar a QoE de certo fluxo de vídeo VF_i , como o introduzido pelo autor Medeiros (MEDEIROS et al., 2018b). A avaliação híbrida de QoE mede o nível de qualidade, em tempo real através das informações dos cabeçalhos de pacotes de *codecs* de IP e de vídeo (GREENGRASS; EVANS; BEGEN, 2009). Assim, uma técnica de Aprendizado de Máquina (*Machine Learning* - ML), como, a Floresta Randômica - (*Random Forest* - RM), prevê o valor do MOS na perda de quadros e nas características de vídeo com baixa complexidade.

No lado do cliente, um monitor de pacotes examina o fluxo MPEG que representa um fluxo constante de *bits* do pacote. Os *bits* precisam ser organizados em pacotes para que erros não se propaguem além das fronteiras de um único pacote. Assim, verifica-se a possibilidade da perda de um tipo de quadro em um GoP g e, se confirmada essa perda, a taxa de quadros perdidos é calculada. Isso ocorre porque, pois a taxa de perda de quadros e o tamanho do GoP afetam de maneira diferente a QoE dos vídeos transmitidos (GREENGRASS; EVANS; BEGEN,

2009). Considera-e, ainda, um processo inteiro de aprendizado de máquina (treinamento, teste e validação) para prever o valor do MOS para um determinado fluxo de vídeo VF .

Nesse sentido, o monitor QoE considera uma RM com técnica de aprendizado de máquina de baixa complexidade, para correlacionar a taxa de perda de quadros I, P e B e o tamanho do GoP com os valores MOS atribuídos, atingindo, portanto, uma pontuação final do MOS. A RM forma seleções menores de uma árvore, informa resultados diferentes nelas e conta a solução mais escolhida, para, assim, responder a pergunta: qual é o valor estimado do MOS considerando o tamanho do GoP e a taxa de perda de quadros I, P e B?

4.1.1.3 QoS e Monitor de Sinal

Em relação aos parâmetros de QoS, o algoritmo Skip-HoVe leva em conta a Taxa de Entrega de Pacotes - (*Packet Delivery Rate* - PDR), que é a proporção entre o número de pacotes de dados entregues e enviados. A PDR reflete a capacidade de processamento da rede e a capacidade de transferência de dados. O protocolo tem um bom desempenho se o valor da PDR for alto quando avalia a conexão entre um veículo v_i e uma célula b_j . Cada veículo v_i calcula o PDR usando IDs de pacotes para detectar os pacotes perdidos e associados à célula b_j . A Equação 4.4 calcula a taxa de entrega de pacotes.

$$PDR = \frac{\sum \text{Pacotes Recebidos}}{\sum \text{Pacotes Enviados}} * 100 \quad (4.4)$$

Da perspectiva do rádio, o algoritmo Skip-HoVe considera o valor RSRQ calculado por um determinado veículo v_i para cada mensagem de *beacon* transmitida por uma célula b_j (células ativas e candidatas). O RSRQ mede a qualidade do sinal recebido nas redes LTE e é calculado de acordo com a Equação 4.5.

$$RSRQ = \frac{N \times RSRP}{RSSI} \quad (4.5)$$

Todas as medições são enviadas para célula de atendimento do veículo e são solicitadas pelo gerenciamento de mobilidade durante a etapa de decisão.

4.1.2 Etapa de Decisão

Nesta etapa, o Gerenciador de *Handover* calcula uma pontuação S_j para cada célula disponível $N(b_j)$ com base na Equação 4.6, a fim de encontrar a próxima célula disponível b_j com melhor QoS e QoE a um determinado veículo v_i se conectar. O algoritmo Skip-HoVe pondera suas técnicas destinadas a diversas métricas com prioridades diferentes para decisões de *handover* e, portanto, precisa atribuir um peso w_i a cada um delas, com a entrada M_i (QoE, QoS

e Distância). Assim, os pesos podem representar quantas vezes a QoE é mais, ou menos crítica que a QoS.

$$S_j = \sum_{i=1}^n w_i \times M_i. \quad (4.6)$$

Em relação ao processo AHP (SAATY, 2005), este tem a finalidade de calcular o fator de influência de cada parâmetro, uma vez que fornece uma técnica estruturada para a tomada de decisões que possuem problemas com vários parâmetros envolvidos. O AHP decompõe um contraponto complexo em uma hierarquia de subproblemas mais simples, combinando, então, fatores qualitativos e quantitativos, para a análise, permitindo que o sistema encontre uma solução ideal, assim que houver vários critérios considerados no processo de entrega.

Especificamente, o AHP abrange uma comparação pareada entre os valores numéricos de cada parâmetro coletado, bem como, seus graus relativos de importância, a fim de ajustar seus pesos em tempo de execução. Um valor numérico representa essa comparação entre pares, pois eles não devem se contradizer. Por exemplo, se uma métrica i (comparação invertida) for duas vezes mais importante que a métrica j , então j será $1/2$ vezes tão importante quanto i . Dessa forma, conforme o exposto, foram definidos cinco níveis de importância entre um parâmetro i e um parâmetro j , como na Tabela 3.

Tabela 3 – Importância do Contexto em Pares

Níveis de Importância		
Seq.	$c_{i,j}$	Definição
1	4	i é muito mais importante que j
2	2	i é mais importante que j
3	1	i é tão importante quanto j
4	$1/2$	i é menos importante que j
5	$1/4$	i é muito menos importante que j

O gerenciador *handover* constrói para cada veículo v_i uma matriz de comparação $A = (C_{i,j})_{m \times m}$, na qual linhas e colunas representam todas as comparações entre pares, de acordo com a Equação 4.7, onde $c_{i,j}$ é uma combinação de importância que o elemento observador i faz do elemento imagem quando comparado com o j e m indica o número de elementos a serem comparados entre si.

$$A = (C_{i,j})_{n \times n} = \begin{pmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,n} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n,1} & c_{n,2} & \cdots & c_{n,n} \end{pmatrix}. \quad (4.7)$$

Ou seja, a matriz $C_{i,j}$ indica quais parâmetros têm maior prioridade uns sobre os outros, conforme a Equação 4.8, onde d é a distância normalizada na qual a QoE representa a pontuação prevista do MOS e o Sinal retrata a intensidade do RSRQ e a QoS representa a PDR. Assim, na primeira linha, observa-se que a métrica da distância d é duas vezes mais importante que a QoE e quatro vezes mais crítica que a QoS e o Sinal. Já a diagonal principal sempre será igual a 1, pois é onde uma métrica é comparada com ela mesma.

$$C_{i,j} = \begin{matrix} & d & QoE & QoS & Sinal \\ \begin{matrix} d \\ QoE \\ QoS \\ Signal \end{matrix} & \begin{pmatrix} 1 & 2 & 4 & 4 \\ 1/2 & 1 & 2 & 2 \\ 1/4 & 1/2 & 1 & 1 \\ 1/4 & 1/2 & 1 & 1 \end{pmatrix} \end{matrix}. \quad (4.8)$$

Ademais, o vetor próprio da matriz ($C_{i,j}$) é calculado dividindo cada elemento pela soma de sua coluna, $W = [0.5 \ 0.25 \ 0.125 \ 0.125]$. Logo, as distâncias normalizadas terão um peso de 0,5 para d , 0,25 para QoE e 0,125 para QoS e Sinal. Analisa-se, também a Taxa de Consistência - (*Consistency Ratio* - RC) e o Índice de Consistência - (*Consistency Index* - CI) do vetor de peso derivado verificando se ele está correto. Nesse sentido, se $CI = 0$ e $CR = CI/RI \leq 0.1$, então a inconsistência da matriz de comparação construída é aceitável. Após o processo de verificação de consistência, descobre-se que a matriz de comparação ($C_{i,j}$) possui $CI = 0$ e $CR = 0$. Logo, a inconsistência da matriz de comparação em pares tornou-se aceitável para atender os critérios de validação definidos para AHP (SAATY, 2005).

O gerenciador de *handover* executa o produto entre o vetor próprio e outro vetor que armazena os valores medidos M_i , obtendo, dessa maneira, a pontuação de S_i destinada a todas as células disponíveis $N(b_j)$. Ou seja, o gerenciador de *handover* seleciona a célula com a pontuação S_i mais alta, sendo, então, a mais adequada para o veículo v_i conectar-se no momento. Na etapa de decisão de *handover*, o gerenciador precisa decidir se uma entrega é necessária, uma vez que sua execução é onerosa e deve ser evitada se não for essencial.

O algoritmo Skip-HoVe, baseia-se em um limite de QoE para acionar o *handover*, o qual é definido com peso 4 para o valor previsto de QoE (LIOTOU et al., 2015). Portanto, assim que o valor previsto de QoE calculado pelo monitor estiver acima desse limite, uma entrega será considerada desnecessária e será, ignorada, pois o vídeo foi entregue ao veículo v_i com uma QoE aceitável. Por outro lado, assim que o *handover* for necessário, o gerenciador escolherá as melhores células disponíveis $N(b_j)$ para o veículo v_i conectar-se.

O Algoritmo 1 possui as principais operações executadas pelo Skip-HoVe para fornecer conteúdo de vídeo utilizando o suporte a QoE em VANET ultradensa. O gerenciador de *handover*

executa todas as três etapas, enquanto o nó móvel está conectado a qualquer célula.

Algoritmo 1: Execução do Algoritmo Skip-HoVe

```

1   $\forall$  Veículos na Rede  $v_i \in V$ 
2  while Veículo está Conectado do
3      O veículo envia medições para sua célula de atendimento Gerenciador Handover
        para iniciar a etapa de decisão.
4      for cada célula disponível  $N(b_i) \in B$  do
5          if QoS está acima de um limite e não está diminuindo then
6              Ignora Handover
7          end
8          else
9              Estimar a próxima posição do veículo
10             Calcular a pontuação  $S_i$  para a célula
11         end
12     end
13     BestCellId  $\leftarrow$  Célula com a mais alta  $S_i$ 
14     if BestCellId  $\neq$  ServingCellId and BestCellRSRQ  $\geq$  Threshold then
15         if Handover é um Ping-Pong then
16             Ignora Handover
17         end
18         else
19             Iniciar a etapa de execução da entrega
20         end
21     end
22 end

```

O algoritmo Skip-HoVe também analisa se a decisão constitui um *handover* com efeito *ping-pong* (Por exemplo, quando um veículo sai de uma célula e retorna em até 4 segundos (TARTARINI et al., 2018)). Nesse caso, o algoritmo Skip-HoVe pula a execução da entrega, sendo considerado um desperdício para os recursos da rede. Por outro lado, o algoritmo, guarda essa célula para decisões de *handover* futuras.

4.1.2.1 Cenário de Previsão de Mobilidade

Considera-se o modelo ARIMA e o modelo KF como técnica de previsão de mobilidade apontada pelo Skip-HoVe. Porém, qualquer outro esquema de previsão de posição pode ser adotado. O ARIMA e o KF preveem a posição futura do veículo $L(x_i, y_i, t + 1)$ com base na atual $L(x_i, y_i, t)$, conforme ilustrado na Figura 10 Nesse sentido, o Skip-HoVe, complementa o algoritmo de previsão de mobilidade toda vez que uma nova medição chega, levando em conta que os intervalos entre as medidas que definem a granularidade do filtro. Nos testes apresentados,

foi adotada a granularidade de 1 segundo.

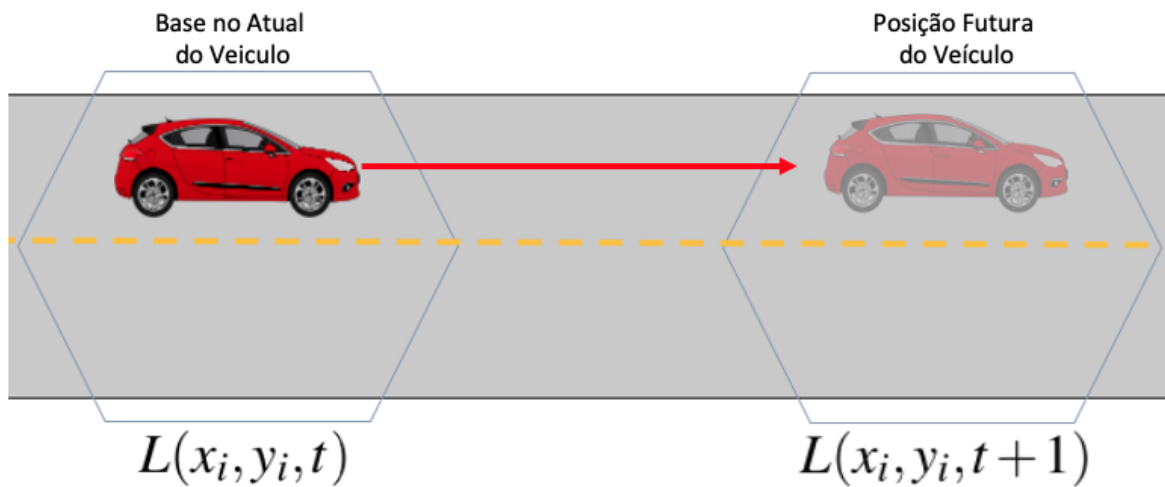


Figura 10 – Medição de Previsão de Mobilidade
Fonte: Autor (2020)

4.1.2.1.1 ARIMA

O Média Móvel Integrada Autorregressiva (*AutoRegressive Integrated Moving Average* - ARIMA) é um modelo estatístico que analisa e prevê séries temporais nas quais os dados processados não são estacionário, pois este modelo pode ser aplicado uma ou mais vezes para eliminar a não estacionariedade quando for necessário. A estrutura, a variância e a média de autocorrelação de uma séria temporal estacionária não mudam no decorrer do tempo, logo, ela não possui tendência. Portanto, a amplitude de suas variações em torno da média é constante. Sendo assim, no modelo ARIMA os, valores futuros das séries são assumidos como uma combinação linear de seus valores e médias móveis passados. O ARIMA é descrito como uma tupla (p, d, q) , na qual p corresponde ao número de medições passadas ponderadas na estimativa, d consiste no número de séries diferenciais que o tornarão estatisticamente estacionário e q corresponde às número de médias móveis anteriores.

A formulação básica do modelo ARIMA é dada pela Equação 4.9, onde, y são os valores passados, ε são as médias móveis anteriores e θ e ϕ são pesos individuais para cada valor treinados pelo modelo.

$$y_t = \theta_0 + \phi_1 y_{t-1} + \phi_2 y_{t-1} + \phi_3 y_{t-3} + \dots + \phi_p y_{t-p} + \varepsilon_0 + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-1} + \theta_3 \varepsilon_{t-3} + \dots + \theta_q \varepsilon_{t-q}. \quad (4.9)$$

O número de valores e médias móveis passados depende das séries temporais estudadas. Algumas delas dependem principalmente dos valores passados ponderados e não precisam médias móveis. O modelo pode ser representado pela Equação 4.9, o que significa que foram

usados cinco termos passados, executando uma diferenciação e não considerando médias móveis passadas. Outrossim, o ARIMA também é usado para prever uma série temporal de variável única e, portanto, deve ser realizada uma etapa de treinamento separadamente, destinada às medições de latitude e longitude. O primeiro passo para a formulação geral do ARIMA consiste em definir a ordem de diferenciação, isto é, o número de vezes que um valor é subtraído do próximo fornecido pelo parâmetro d , de acordo com a Equação 4.10. O modelo ARIMA pode ser usado para a previsão de mobilidade do veículo $L(x_i, y_i, t + 1)$. Nesse sentido, deve ser treinado individualmente, em relação aos veículos e, também, unicamente a cada coordenada (latitude e longitude).

$$y_t = \begin{cases} Y_t, & \text{if } d = 0 \\ (Y_t - Y_{t-1}), & \text{if } d = 1 \\ (Y_t - Y_{t-1}) - (Y_{t-1} - Y_{t-2}), & \text{if } d = 2 \\ \text{continuamente} & \infty \end{cases} \quad (4.10)$$

4.1.2.1.2 Filtro de Kalman

O Filtro de Kalman - (*Kalman Filter* - KF), possui a finalidade de tentar estimar um estado $x_t \in \mathbb{R}^n$ com base no estado anterior x_{t-1} , isto é, o filtro precisa apenas do valor do estado anterior, para estimar o próximo. O estado x em um KF consiste em o vetor que contém um par de coordenadas geográficas do veículo g_t , ou seja, latitude e longitude, em certo momento t (Exemplo: $L(x_i, y_i, t)$).

Explicitamente, o processo é modelado por uma equação de diferença estocástica, representada pela Equação 4.11, onde A é uma matriz $n \times n$ que relaciona o estado anterior ao atual e $w \in \mathbb{R}^n$, a fim de estimar o ruído.

$$x_t = Ax_{t-1} + w_{t-1}. \quad (4.11)$$

A estimativa leva em conta uma medida dada por Z_k , calculada pela Equação 4.12. Então, ela pode ser modelada em termos da previsão com um fator de correção H e ruído v_k .

$$Z_k = Hx_k + v_k. \quad (4.12)$$

Defini-se \hat{x}_k^- como o estado anterior, x_k como estado previsto e \hat{x}_k como estado seguinte, sendo que \hat{x}_k^- e \hat{x}_k são valores reais do processo. Também, estima-se x_k com base na medida Z_k . Os erros anteriores e seguintes são definidos por e_k^- e e_k , e calculados pelas Equações 4.13 e 4.14, respectivamente.

$$e_k^- = x_k - \hat{x}_k^-. \quad (4.13)$$

$$e_k = x_k - \hat{x}_k. \quad (4.14)$$

Ademais, a covariância do estado anterior, é definida com base na Equação 4.15 e a covariância do estado seguinte pela Equação 4.16, onde o valor esperado do erro é e_k^t é multiplicado pela matriz transposta de erros. O objetivo do filtro é minimizar a covariância de erro P_k .

$$P_k^- = E [e_k^- e_k^{-T}]. \quad (4.15)$$

$$P_k = E [e_k e_k^T]. \quad (4.16)$$

Logo, o estado seguinte é expresso como uma combinação linear do estado anterior e um termo de correção proporcional à diferença entre o valor do estado e o valor medido, conforme a Equação 4.17. Assim, o valor de \hat{x}_k corresponde ao vetor de coordenadas previstas na próxima medição g_{t+1} .

$$\hat{x}_k = \hat{x}_k^- + K (z_k - H\hat{x}_k^-). \quad (4.17)$$

A matriz K $n \times m$ é o ganho que tem como objetivo, minimizar a covariância de erro. É possível minimizar o erro substituindo a Equação 4.18 na Equação 4.19 e, em seguida, derivando o resultado. Essas equações são, as fórmulas finais para calcular o ganho do filtro a ser usado na estimativa.

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1}. \quad (4.18)$$

$$K_k = \frac{P_k^- H^T}{H P_k^- H^T + R}. \quad (4.19)$$

4.1.2.2 Precisão da Previsão de Mobilidade

A previsão de mobilidade do KF e do ARIMA, foi testada por meio de um conjunto de dados veiculares, a fim de escolher um deles para fazer parte do algoritmo de *handover*. Nessa perspectiva, considera-se um *trace* de mobilidade veicular real coletado em aproximadamente 500 táxis da cidade de São Francisco (USA)(PIORKOWSKI; SARAFIJANOVIC-DJUKIC; GROSSGLAUSER, 2009).

O conjunto de dados consiste em medições de dados extraídos de dispositivos GPS de mais de 500 táxis na área da baía de São Francisco nos Estados Unidos durante um período de um mês, gerando mais de 10 milhões de amostras. O modelo ARIMA é aplicado nesse conjunto

de dados, considerando dois valores passados e uma média móvel. A série temporal é desviada duas vezes para se tornar estacionária. Esses parâmetros foram encontrados, através do uso de um estimador de *Grid Search*¹, visando o melhor desempenho. 60% dos dados foram usados para treinamento e 40% para testes.

A Figura 11 mostra o Desvio da Raiz Quadrada da Média - (*Root-Mean-Square Deviation* - RMSE)² para o ARIMA e ao KF, provendo a localização de cada veículo no conjunto de dados. Analisando os resultados, observa-se que o KF apresenta um erro 85,7% maior que o ARIMA.

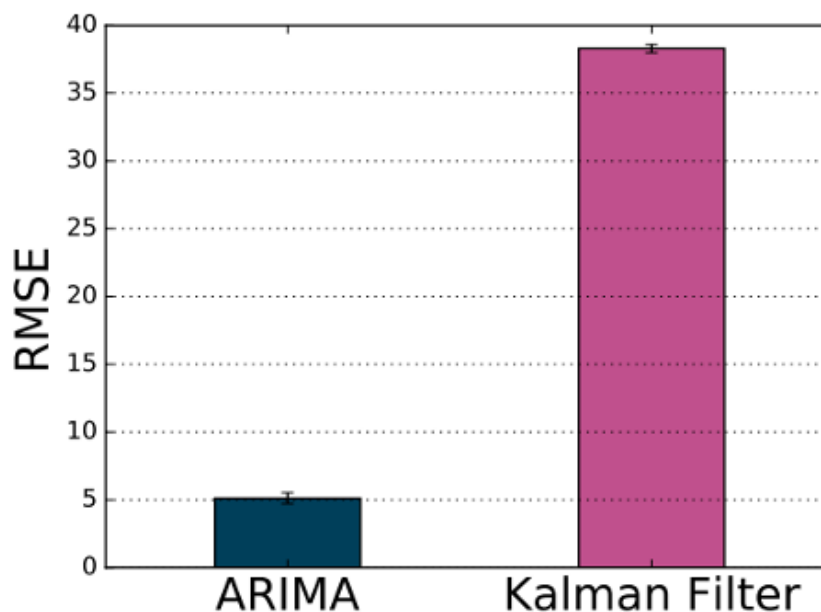


Figura 11 – RMSE para o ARIMA e para o filtro Kalman
Fonte: Autor (2020)

O movimento dos veículos tende a ser irregular e não linear na maior parte do tempo, contudo o KF é mais preciso quando os dados analisados têm natureza linear, e interativa. Nesse seguimento, o KF precisa de tempo para se ajustar às mudanças de mobilidade em parâmetros como velocidade e direção, de forma isto é, os ajustes se dão de forma *on-line*. Por outro lado, o ARIMA pode prever o padrão de mobilidade com alta precisão, mesmo após ser treinado. Ele é muito robusto, apesar de possuir dados não lineares. Os resultados do RMSE podem ser explicados por meio da Figura 12, a qual mostra a longitude do veículo ao longo do tempo.

Ao analisar os resultados, conclui-se que as previsões do ARIMA estão muito mais próximas dos pontos de dados originais, enquanto que as previsões do KF, em alguns casos, encontram-se demasiadamente distantes dos pontos de dados originais. Logo, na amostra 30, o veículo virou para a esquerda ou para direita. O ARIMA consegue prever esse padrão de mobilidade, enquanto o KF não o detecta.

¹ A *Grid Search* é usada para encontrar os hiperparâmetros ótimos de um modelo que resultam nas previsões mais "precisas".

² O RMSE é o desvio da média quadrática da raiz ou o erro quadrático médio da raiz. É uma medida frequentemente usada das diferenças entre os valores previstos por um modelo ou estimador e os valores observados.

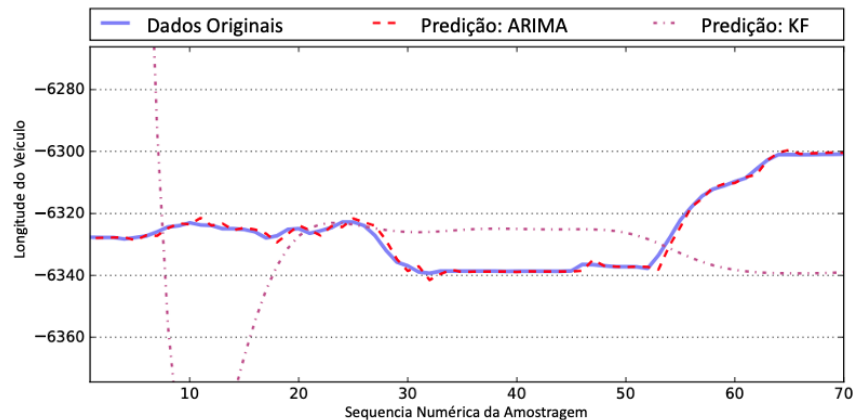


Figura 12 – As previsões de longitude do veículo para o ARIMA e KF.

Fonte: Autor (2020)

4.2 Conclusão do Capítulo

Este Capítulo apresentou um algoritmo de *handover*, que possui diversos critérios para distribuição de vídeo em cenários de VANET ultradensa, chamado Skip-HoVe. Também foi abordado detalhadamente o modelo aplicado na rede e seus sistemas, as etapas de medições que coletam os dados dos veículos e das células disponíveis, assim como a predição de mobilidade, o monitor de sinal e a QoE e suas etapas de decisões. Assim, conclui-se que este Capítulo foi embasado no algoritmo Skip-HoVe que incorpora todos os recursos críticos expostos e discutidos, os quais não haviam sido fornecidos em um algoritmo de *handover* unificado previamente.

5 AVALIAÇÃO DO ALGORÍTIMO SKIP-HOVE

Este descreve a metodologia de avaliação, incluindo a descrição do cenário, os parâmetros de simulação e as métricas usadas, no sentido de avaliar o desempenho de diferentes algoritmos de *handover*, destinados a distribuição de vídeo em um cenário de VANET ultradensa.

5.1 Descrição e metodologia do cenário

O algoritmo Skip HoVe foi implementado na arquitetura do *Network simulator* (NS), versão 3.29¹ e está disponível para download no Github². Cabe registrar que na respectiva versão está implementada a pilha de protocolos LTE para comunicação V2V e V2I. Um cenário de VANET ultradensa foi considerada na simulação, de acordo com Demarchou (DEMARCHOU; PSOMAS; KRIKIDIS, 2018) e 3GPP LTE release 13 (ACCESS, 2013), observando uma área de 2km x 2km com 7 macrocélulas que cobrem todo o cenário até certo ponto e 100 células pequenas distribuídas por toda área. Os principais parâmetros de simulação são vistos na Tabela 4.

Tabela 4 – Parâmetros de Simulação Aplicados

Detalhes dos Parâmetros usados na Simulação		
#	Parâmetros	Valor
01	Número de veículos	[50, 100, 150, 200]
02	Velocidade média dos veículos	43.81 km/h
03	Número de macrocélulas	7
04	Número de células pequenas	100
05	Potência de transmissão das macrocélulas	46 dBm
06	Potência de transmissão das células pequenas	23 dBm
07	Altura das células pequenas	10 Metros
08	Altura das macrocélulas	45 Metros
09	Modelo de perda de propagação	Nakagami
10	Tamanho do cenário	2km × 2km
11	Sequência de vídeo testada	Highway
12	Frequência <i>Downlink</i>	2120 (MHz)
13	Frequência <i>Uplink</i>	1930 (MHz)

As macrocélulas possuem uma potência de transmissão de 46dBm, enquanto as células pequenas, apresentam uma potência de transmissão de 23dBm. A simulação fundamenta-se através do modelo de propagação Nakagami³, o qual pode ser adequado a diferentes cenários urbanos (RUBIO; REIG; CARDONA, 2007). Foram realizadas 33 simulações geradas alea-

¹ <http://www.nsnam.org/>

² <https://github.com/allandoug/>

³ Nakagami-m, é utilizado para a derivação da distribuição de probabilidade relacionada *gamma* e direcionada para meios de radio comunicação.

toriamente através do gerador de números pseudo-aleatórios do simulador (MRG32k3a)⁴. Os resultados, demonstram os valores com um intervalo de confiança de 95%.

Foi empregado o rastreamento de mobilidade de táxis de São Francisco (PIORKOWSKI; SARAFIJANOVIC-DJUKIC; GROSSGLAUSER, 2009) para a simulação de mobilidade de tráfego e veículo, conforme descrito na Seção 4.1.2.2, variando o número de veículos entre 50, 100, 150, e 200 para avaliar a escalabilidade.

Observa-se o cenário real representado pelos dados coletados no *trace*, ou seja, a área da Baía de São Francisco, nos EUA, devido à sua relação direta com o mundo real, tal como os padrões de mobilidade humana. A Figura 13 ilustra o cenário de simulação com diversas células pequenas ao redor de 7 macrocélulas.

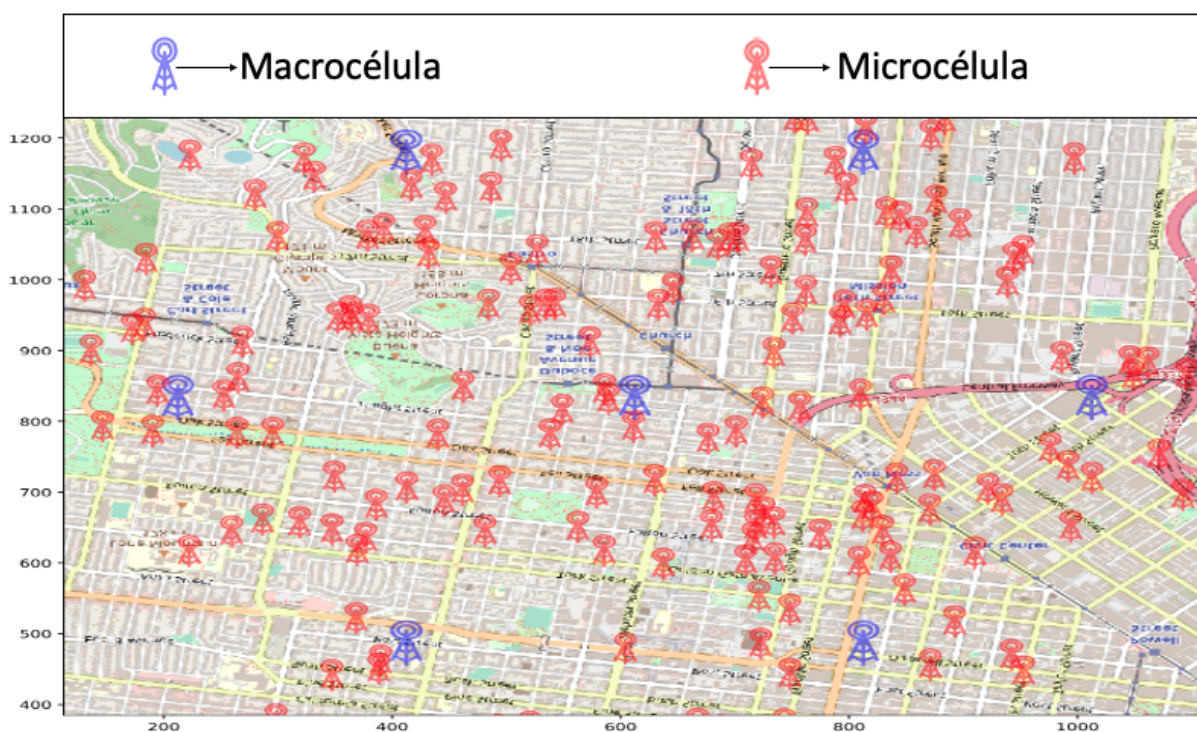


Figura 13 – Amostra do Cenário de Simulação.

Fonte: Autor (2020)

Nesse contexto, a área de cobertura de células pequenas leva a um mosaico de Voronoi⁵ (BACCELLI; BŁASZCZYSZYN, 2001). Assume-se que pelo menos uma macrocélula esteja disponível em todos os pontos do cenário, tal como esperado em ambientes de veículos conectados. Vale-se ressaltar que, um veículo viajando com velocidade média igual a 43km/h cruzou a área de cobertura de 42,8 células pequenas durante as simulações.

Utilizou-se nas simulações um vídeo, chamado *Highway*, com diferentes níveis de complexidade moderada em termos de movimento e de espaço, que pode ser encontrado em

⁴ O MRG32k3a é um gerador aleatório recursivo de múltiplos de fluxos de dados.

⁵ Voronoi é um tipo especial de decomposição de um dado espaço, por exemplo, um espaço métrico, determinado pela distância para uma determinada família de objetos (subconjuntos) no espaço.

um conhecido repositório de vídeo (MONTGOMERY et al., 2020). O vídeo tem duração de 66 segundos codificados com H.264, 30fps é tamanho intermediário de 352×288 pixels e uma taxa de bits de 210kbps. Todos os vídeos avaliados são transmitidos em um *loop*. O decodificador, faz o uso de um método de cópia de quadro como ocultação de erro, substituindo cada quadro perdido pelo último recebido, a fim de reduzir a perda de quadros e manter a qualidade do vídeo.

As simulações foram executadas com cinco algoritmos de *handover* diferentes, descritos seguir:

1. **Algoritmo de *Handover* baseado em SINR:** É o mais comum e tradicional. Ele considera apenas a intensidade do sinal para a decisão de *handover*. O *handover* ocorre assim que houver uma célula de rádio com um valor de intensidade de sinal mais alto que o atual (KURJENNIEMI; HENTTONEN; KAIKKONEN, 2008);
2. **Algoritmo *Handover* PBGT:** Também conhecido como algoritmo de célula mais forte. Ele, executa uma decisão com base na quantidade de energia, onde o *handover* é executado se uma célula vizinha tiver potência recebida superior à da célula servida mais um valor de interesse. . Tal diferença é mantida ao longo de um TTT previamente definido (3GPP, 2011);
3. **Algoritmo de *Handover* NC-Skipping:** Considera a mobilidade para um *handover* não-cooperativo (DEMARCHOU; PSOMAS; KRIKIDIS, 2018);
4. **Algoritmo de *Handover* SER:** Demonstra informações de QoS e QoE para a decisão de *handover* (MEDEIROS et al., 2018b);
5. **Algoritmo Skip-HoVe:** Argumenta vários critérios para a tomada de decisões. É um algoritmo de entrega aprimorado baseado em *handover*, com o objetivo de fornecer mobilidade perfeita para distribuição de vídeo, sem efeitos *ping-pong*, como descrito na no Capítulo 4.

As métricas de QoE superam as limitações das métricas de QoS para avaliação da qualidade do vídeo, uma vez que as métricas de QoS falham ao capturar aspectos subjetivos do conteúdo apresentado no vídeo e relacionados à experiência humana (ARGYRIOU et al., 2017). Desta forma, o nível de qualidade de cada vídeo transmitido é medido através de métricas de QoE objetivas e subjetivas conhecidas, tal como a SSIM e o MOS, respectivamente. Especificamente, a SSIM compara a variação entre o vídeo original e a sequência principal em relação à luminância, ao contraste e à similaridade estrutural. Os valores do SSIM variam de 0 a 1, sendo 0 o pior caso e 1 o vídeo transmitido com a mesma qualidade que a do vídeo original. Considera-se, ainda, o uso do VQMT para medir os valores SSIM de cada vídeo transmitido.

A avaliação subjetiva captura todos os detalhes que podem afetar a experiência do usuário. Nesse contexto, o MOS é uma das métricas mais usadas para avaliação subjetiva, além

de requerer que os avaliadores humanos classifiquem a qualidade geral do vídeo. Além disso, para a avaliação do MOS, o método de estímulo único é usado seguindo as recomendações ITU-R BT.500-11, uma vez que se encaixa muito bem em um grande número de aplicativos multimídia emergentes (ZHENG et al., 2015). Os avaliadores humanos assistem apenas uma vez ao vídeo e, em seguida, fornecem uma pontuação fazendo o uso da escala de qualidade numérica de dez graus, tal como apresentado no Capítulo 2.

Ademais, na avaliação subjetiva, 31 observadores avaliaram os vídeos, incluindo estudantes de graduação e pós-graduação, além de funcionários da Universidade Federal do Estado do Pará. Eles tinham a visão normal e as idades variavam de 18 a 45 anos. Os vídeos distorcidos foram reproduzidos através de um dispositivo Samsung Galaxy Tab A 8.0 com uma tela de 8 polegadas colocada no banco traseiro do encosto de cabeça de um carro. Os vídeos foram avaliados para que as pessoas definissem/pontuassem seus valores de MOS durante viagens entre 9h e 18h. O comportamento humano ao avaliar os vídeos, bem como, as distrações causadas pelo ambiente circundante e, ainda, quaisquer outros fatores psicológicos (subjetivos) relacionados à psicologia humana estão fora do escopo desta tese (FELICE et al., 2015).

Avaliou-se a eficácia da entrega, tendo em vista que toda entrega é um processo oneroso para o ponto de vista da infraestrutura. Dessa forma, uma entrega deve ser executada com cuidado pelo gerenciador, a fim de evitar o desperdício de recursos limitados. Foram consideradas, duas métricas para avaliar a decisão de *handover*, quando o mesmo não é necessário. O número de *handovers* é vital, para fornecer detalhes sobre os tempos médios em que um algoritmo de gerenciamento de *handover* específico, suporta um único usuário móvel, para, assim, alterar sua célula. Além disso, o efeito *ping-pong* é uma métrica importante para avaliar *handovers* desnecessários, dado que ele acontece quando o gerenciador de entrega aciona o dispositivo móvel para executar um *handover* até uma célula. No entanto, alguns segundos depois (4 a 6 segundos), o dispositivo móvel retorna à célula conectada anteriormente executando um segundo *handover*.

5.2 Resultados das Simulações

A Figura 14 mostra os resultados da avaliação objetiva da qualidade do vídeo, considerando os valores de SSIM para um vídeo transmitido por diferentes algoritmos de *handover* (Skip-HoVe, NC-Skipping, SER, SINR-based, e PGBT). Ao analisar os resultados, pode-se concluir que os vídeos fornecidos pelo Skip-HoVe têm consistentemente um valor de SSIM próximo a um, independentemente do número de veículos, o que não é alcançado pelos demais algoritmos de entrega. Por exemplo, o Skip-HoVe entregou vídeos com SSIM 28%, 26%, 27% e 30% superior à dos algoritmos NC-Skipping, SER, SINR e PGBT, respectivamente. Tal razão se dá, pelo fato de o Skip-HoVe fornecer decisões de *handover* sem falhas e de forma confiável em um cenário de VANET ultradensa.

Para esse fim, o algoritmo Skip-HoVe, executa um mecanismo de *handover* associado à previsão de mobilidade e às decisões com reconhecimento de QoS e QoE, o que significa que os *handover* são feitos com mais confiabilidade e menos frequência. Nesse sentido, o algoritmo conseguiu reduzir a taxa de perda de quadros I e o número de *handovers* entre as células, conforme mostrado na Figura 14. No entanto, os demais algoritmos de *handover* não possuem, acabam por não possuir, ao menos uma, de tais características.

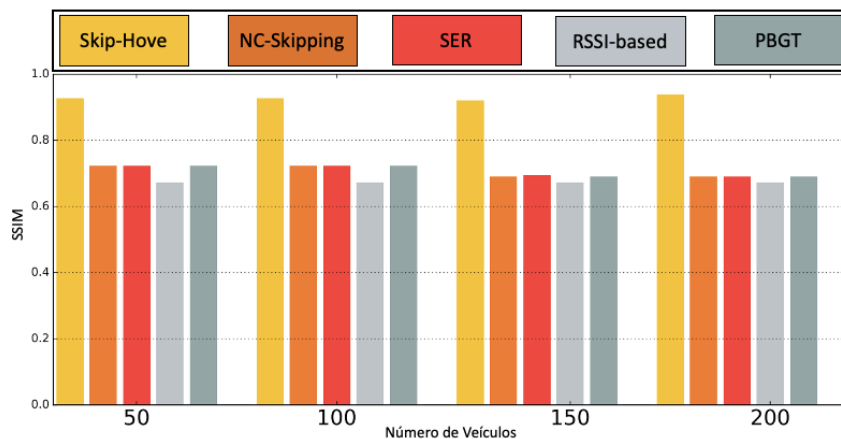


Figura 14 – SSIM para vídeos entregues por diferentes algoritmos de *handover*

A Figura 15 mostra a avaliação subjetiva da qualidade do vídeo, usando a métrica MOS, da qualidade de um conjunto de vídeos transmitido por diferentes algoritmos de *handover*, disponível publicamente no repositório do YouTube⁶.

Analisando os resultados do MOS, conclui-se que o algoritmo Skip-HoVe entregou vídeos em tempo real com perdas de quadros com qualidade acima de regular (valor MOS de 5 e 4). Ao mesmo tempo, os demais algoritmos de *handover* entregaram vídeos com perdas de quadros entre regular e péssimo (valor MOS variando entre 1 e 3).

Isso ocorre, porque cenários de VANET ultradensa acarretam *handover* frequentes e efeito *ping-pong*, aumentando, assim, as perdas de pacotes, especialmente de quadros de vídeo mais importantes, resultando em um MOS ruim. Nesse contexto, o algoritmo selecionou uma célula candidata confiável, direcionada a conexão de um veículo e, portanto, baixou o conteúdo do vídeo considerando várias métricas associadas a uma decisão de entrega baseada em *handover*. Por outro lado, os demais algoritmos como o NC-Skipping, o SER, SINR-based e o PGBT não consideram decisões baseadas em *handover* eficientemente, junto com informações de QoE e QoS. Portanto, os resultados do MOS apresentam melhorias significativas no nível de qualidade do vídeo entregue, utilizando o algoritmo Skip-HoVe, comparado com os dos outros algoritmos de *handover*.

A Figura 16, apresenta a taxa de perda de Quadros I dos vídeos entregues pelos algoritmos de *handover* considerados, os quais ajudam a explicar os resultados de QoE. A disseminação de

⁶ Vídeos das avaliações MOS usados na transmissão em VANETs - <http://bit.ly/3aorpoG>

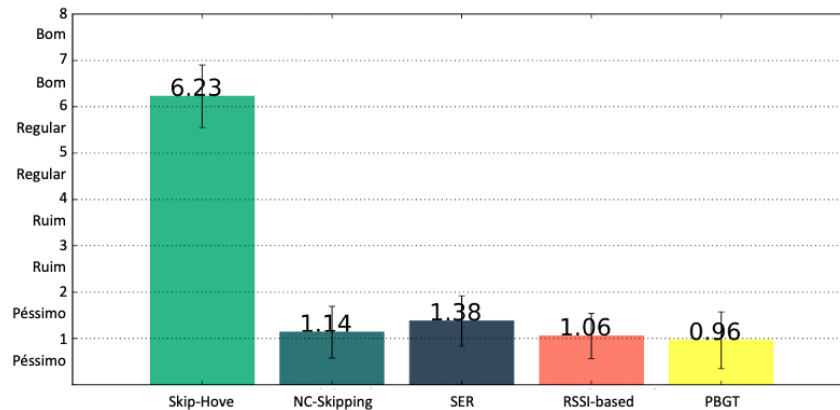


Figura 15 – MOS para vídeos entregues por diferentes algoritmos de *handover*

vídeo em tempo real requer baixa perda de quadro, especialmente dos quadros de vídeo mais importantes, ou seja, quadros I, para que haja a possibilidade de suportar a disseminação de vídeo com suporte a QoE (ARGYRIOU et al., 2017). A perda de um quadro I causa distorções graves de vídeo com base na perspectiva do usuário, uma vez que a qualidade do vídeo é recuperada apenas quando o decodificador recebe um quadro I intacto. Com base nos resultados da simulação, conclui-se que Skip-HoVe reduziu as perdas de quadros I em aproximadamente 94% em comparação com demais algoritmos de *handover*. O algoritmo transmitiu quadros prioritários com alta probabilidade de entrega, comparados com os dos outros algoritmos de *handover* avaliados, aumentando o nível de qualidade do vídeo. Por outro lado, os demais algoritmos entregaram quadros I com taxa de perda que varia de 60% a 80%, independentemente do número de veículos no cenário, fazendo com que o vídeo gaste mais tempo para recuperar a QoE.

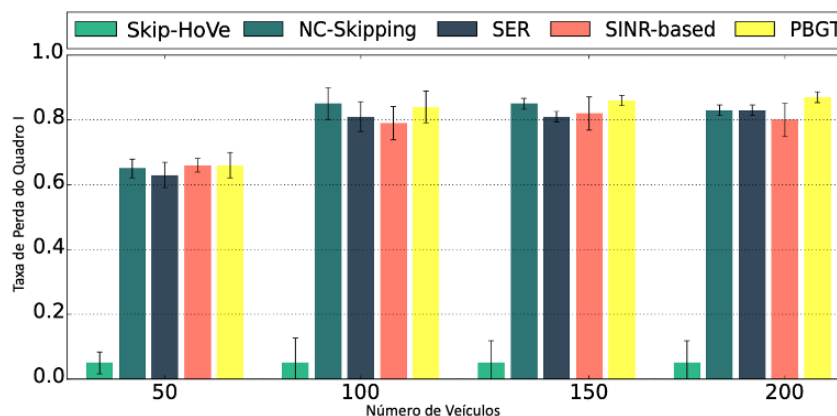


Figura 16 – Taxa de perda de Quadros I para vídeos entregues por diferentes algoritmos de *handover*

Foram selecionados quadros aleatórios (por exemplo, o 683º quadro) da sequência do vídeo *Highway*, possuindo valores de MOS mais altos conforme pode-se notar nas Figuras 17 (b),(d),(e),(f) e (g) e mais baixos representado pelas Figuras 17 (c),(h),(i),(j) e (k) para cada algoritmo de *handover*, a fim de demonstrar o impacto da decisão de entrega executada e

fundamentada na perspectiva do usuário, como mostra a representação geral da Figura 17.

Especificamente, o 683º quadro da sequência do vídeo da rodovia é um quadro P recuperado por uma câmera em um carro transitando em uma rodovia com outro carro preto na faixa esquerda da mesma, como se observa na Figura 17 (a).

Nos dois casos, vídeos com valores MOS mais altos e mais baixos são apresentados. Logo, o quadro entregue pelo algoritmo Skip-HoVe tem a mesma qualidade do quadro original (Figura 17 (b) e Figura 17 (g)), a qual mostra claramente os benefícios do algoritmo Skip-HoVe para a entrega de vídeo em um cenário de VANET ultradensa. Em contraponto, o 683º quadro do vídeo com o valor MOS mais alto, transmitido pelos demais algoritmos de *handover* apresenta distorções mínimas comparado com o quadro original. Entretanto, o carro preto na faixa esquerda não aparece, pois esse quadro foi perdido e reconstruído com base no que foi recebido anteriormente. O algoritmos de *handover* Skip-HoVe, NC-Skipping, SER, SINR e PGBT entregaram o vídeo com uma taxa de perda de quadros I de 4,48%, 97,03%, 98,51%, 13,43% e 97,01%, respectivamente.

Por fim, para o vídeo com valor MOS mais baixo, o quadro transmitido pelos algoritmos de *handover* NC-Skipping, SER, SINR e PGBT, ficou muito prejudicado, comparando com o quadro original,

pois ele foi perdido e ficou impossível reconstruí-lo com quadros recebidos anteriormente, impedindo a visualização com qualidade em sua imagem. Portanto, o vídeo com valor MOS mais baixo fornecido pelos algoritmos de *handover* Skip-HoVe, NC-Skipping, SER, SINR e PGBT apresentou uma taxa de perda de quadros I de 19,4%, 100%, 100%, 16,19% e 100%, respectivamente.

Observe que, mesmo nos melhores casos, os vídeos geralmente não são classificados com a melhor pontuação. Isso ocorre porque a resolução do vídeo original já encontra-se limitada. Como mencionado anteriormente, a ausência de um quadro I, acarreta graves distorções de vídeo, fundamentadas, assim, na perspectiva do usuário, uma vez que a QoE recupera-se apenas quando o decodificador recebe um quadro I intacto. Como os quadros I contêm a maior quantidade de informações para a imagem e, como os vídeos foram reconstruídos usando o método de cópia de quadro, os algoritmos NC-Skipping, SER e PGBT apenas reconstroem a imagem com as informações que mudam de um quadro para o outro, diminuindo a QoE. Portanto, pode-se concluir que o Skip-HoVe apresenta um bom desempenho, visando fornecer vídeos com a excelente qualidade, comparado com os demais algoritmos de *handover* analisados.

As Figuras 18 e 19 exibem o resultado da SSIM de cada quadro que compõe as sequências do vídeo da rodovia transmitidas pelos algoritmos de *handover* avaliados, ajudando a explicar os resultados da Figura 17. Ao analisar os resultados, observa-se que o algoritmo Skip-HoVe forneceu decisões de *handover* rápidas e confiáveis para os veículos baixarem o vídeo nos dois casos, isto é, o algoritmo Skip-HoVe entregou quase todos os quadros com SSIM próxima a 1

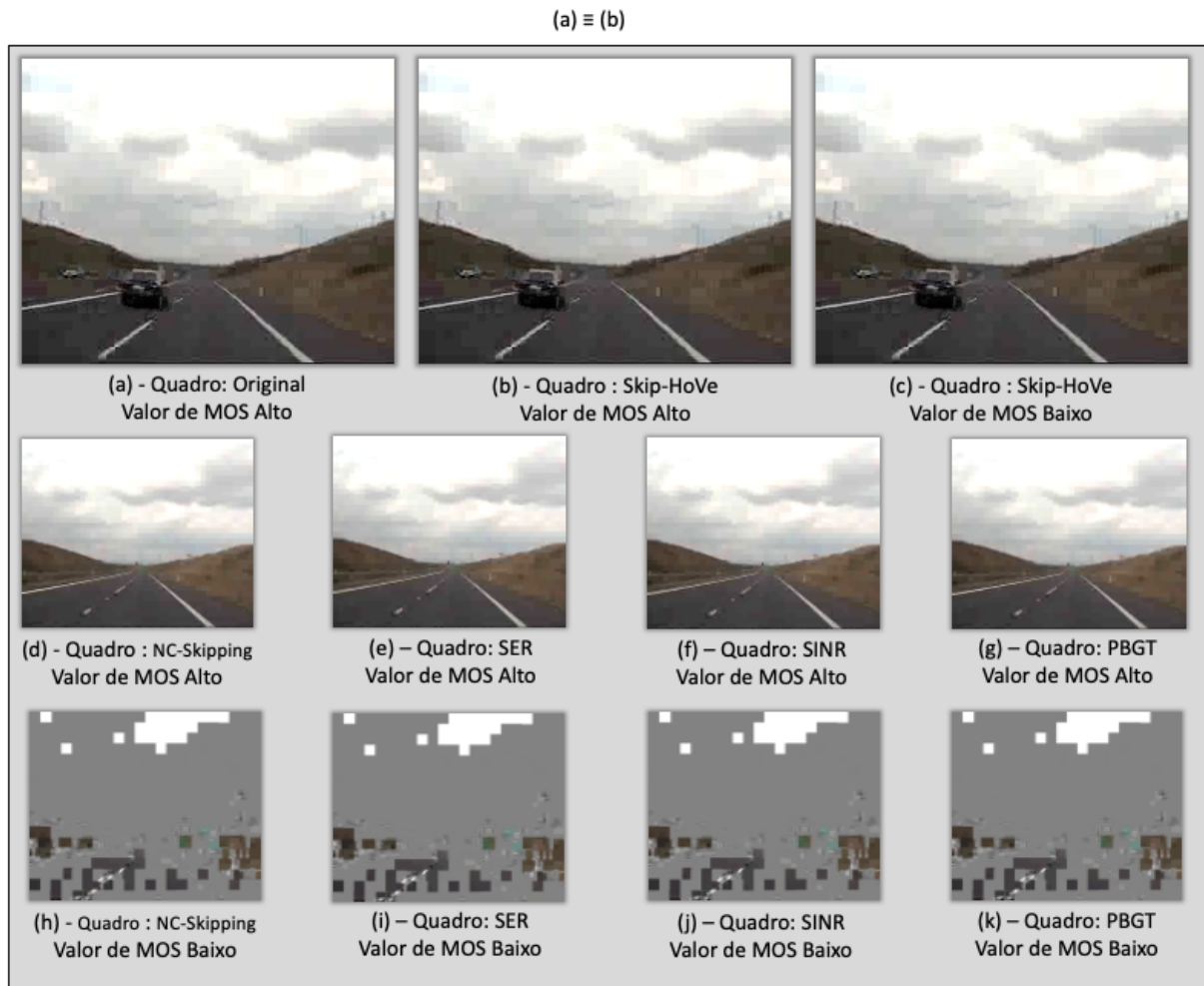


Figura 17 – Comparativo do 683º quadro do vídeo transmitido por diferentes algoritmos de *handover*

e acima de 0,8 . Para o vídeo com o valor MOS mais alto, os demais algoritmos de *handover* começaram com uma conexão ruim ($SSIM < 0,7$). Após algumas decisões de *handover*, esses algoritmos aumentaram a $SSIM$ para 0,9, porém ela diminuiu posteriormente. Finalmente, os algoritmos de *handover* entregaram os quadros com $SSIM$ de 0,3 a 0,7 para o vídeo com o valor MOS mais baixo. Isso ocorre porque estes algoritmos experimentam muitos *handover*, especialmente com o efeito *ping-pong* que piora a QoE dos vídeos entregues.

Para o gráfico da Figura 18, os algoritmos Skip-HoVe, NC-Skipping, SER, SINR-based e PBGT tiveram resultados de 25, 7, 35, 25 e 2 em suas *handovers*, sendo 5, 2, 13, 7 e 0 consideradas *handover* com o efeito *ping-pong*, respectivamente. Além disso, os algoritmos já existentes não consideram várias métricas associadas à uma decisão de entrega baseada em *handover* executá-las confiavelmente. Isto posto, observa-se que o algoritmo Skip-HoVe é o único capaz de fornecer uma experiência perfeita, sem quedas de QoE nos cenários avaliados, fornecendo os pacotes essenciais e garantindo alta fidelidade à sequência de vídeo original.

A Figura 20 apresenta o número de *handovers* executados durante a simulação por cada algoritmo. Observa-se que ao realizar a menor quantidade de *handover* não é necessariamente a

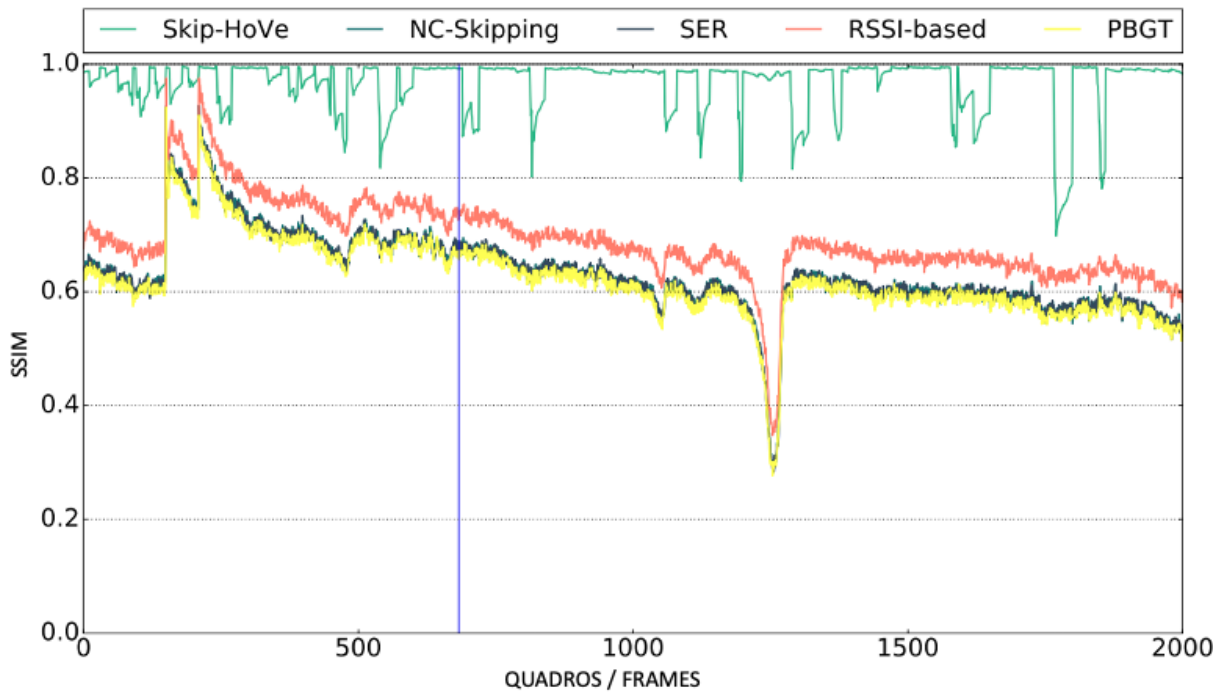


Figura 18 – Valor MOS - Alto

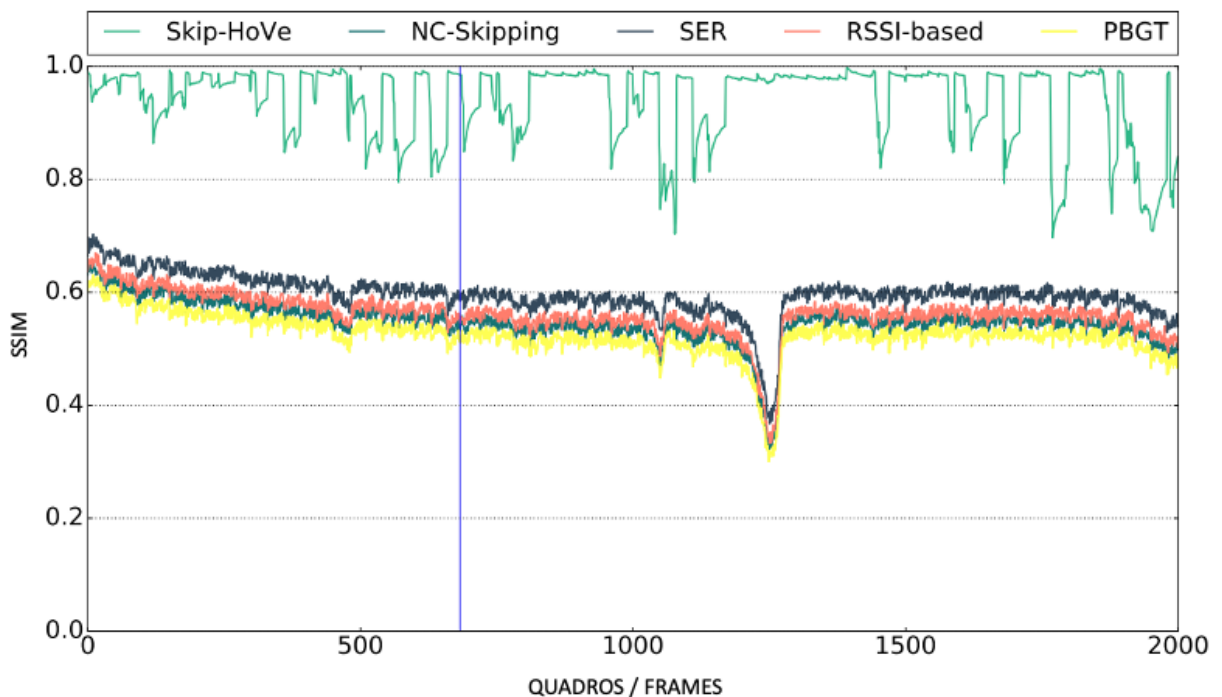


Figura 19 – Valor MOS - Baixo

melhor abordagem, pois o algoritmo PBGT fornece vídeos com QoE ruim sem quase realizar *handover*. Em contrapartida, a técnica de *handover* empregada pelo NC-Skipping é ineficiente em fornecer QoE aceitável, sem uma decisão de vários critérios. O SER tem uma decisão de *handover* com reconhecimento de QoE, porém, não considera o esquema baseado em *handover*, acumulando, assim, em do alto número desse processo. O SINR, por outro lado, efetua menos *handover* que Skip-HoVe, mas é altamente suscetível à ocorrência do efeito *ping-pong*. Curiosa-

mente, isso faz com que o SINR, obtenha resultados semelhantes aos do SER, mostrando, então, o impacto significativo de *handover* frequentes, mesmo na decisão baseada em QoE do SER. Finalmente, pode-se ver que, embora o NC-Skipping e o Skip-HoVe realizem aproximadamente a mesma quantidade de *handover*, o NC-Skipping falha ao fornecer níveis aceitáveis de QoE e a QoS.

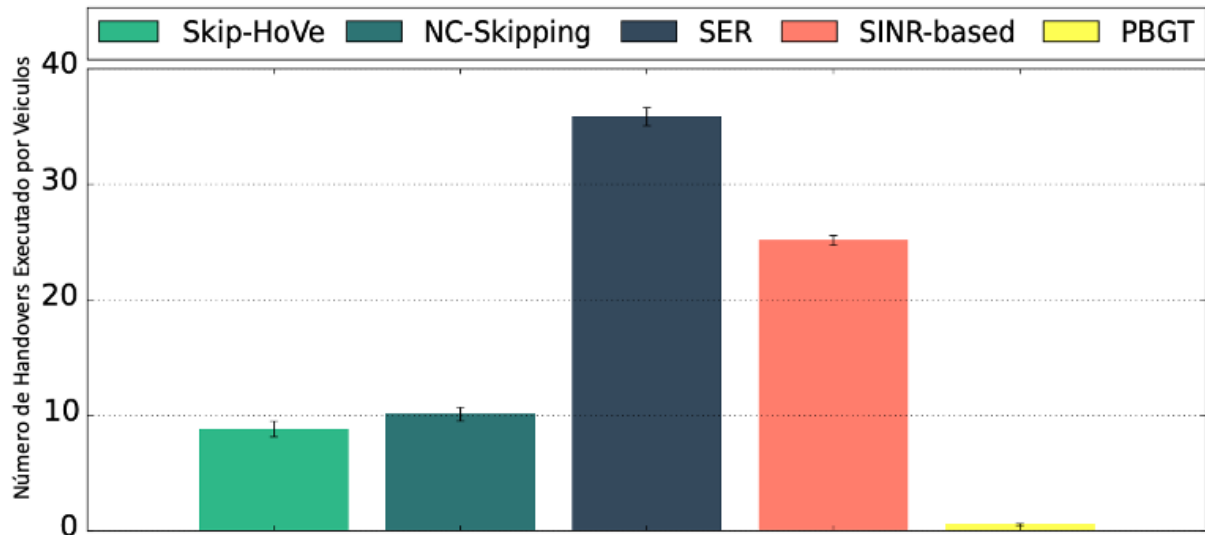


Figura 20 – Número de *handover* executadas por diferentes algoritmos de *handover*.

A Figura 21 mostra a taxa de *handover* com efeito *ping-pong*, para os algoritmos de Skip-HoVe, NC-Skipping, SER, SINR-based e PBGT. É essencial destacar que o *handover* com o efeito *ping-pong*, é considerado quando o usuário sai de uma célula e retorna para dentro dela depois de 4 segundos. Analisando os resultados, conclui-se que o algoritmo Skip-HoVe mantém a taxa de *ping-pong* em torno de 2% devido à sua ótima política decisória de conseguir evitar este efeito. Como mencionado anteriormente, o algoritmo PBGT realiza uma quantidade menor de *handovers* e, conseqüentemente, apresenta menor probabilidade de acontecer o efeito *ping-pong* dentro da janela considerada. Em contrapartida,

este efeito acontece mais frequentemente com os algoritmos NC-Skipping, SER e SINR-based por estes não terem uma barreira transparente contra ele. Mesmo com um mecanismo de *handover*, esses algoritmos não estão associados a uma estratégia de múltiplos critérios, tornando-se suscetíveis ao efeito *ping-pong*.

A Tabela 5 resume os resultados experimentais obtidos, por cada um dos algoritmos de *handover* para o caso de um vídeo mostrado aleatoriamente. Observa-se que, em termos de QoE, o Skip-HoVe obteve a pontuação MOS mais alta entre todos os algoritmos avaliados, seguido por SINR-based, PBGT, SER e NC-Skipping, respectivamente. O algoritmo Skip-HoVe executa decisão baseada em QoE, porém apenas ele suporta as particularidades de redes ultradensas, tais como a grande quantidade de células. Em média, o Skip-HoVe realizou menos *handover*, que os demais algoritmos, com exceção do PBGT, no qual os veículos saíram apenas de uma macrocélula para outra.

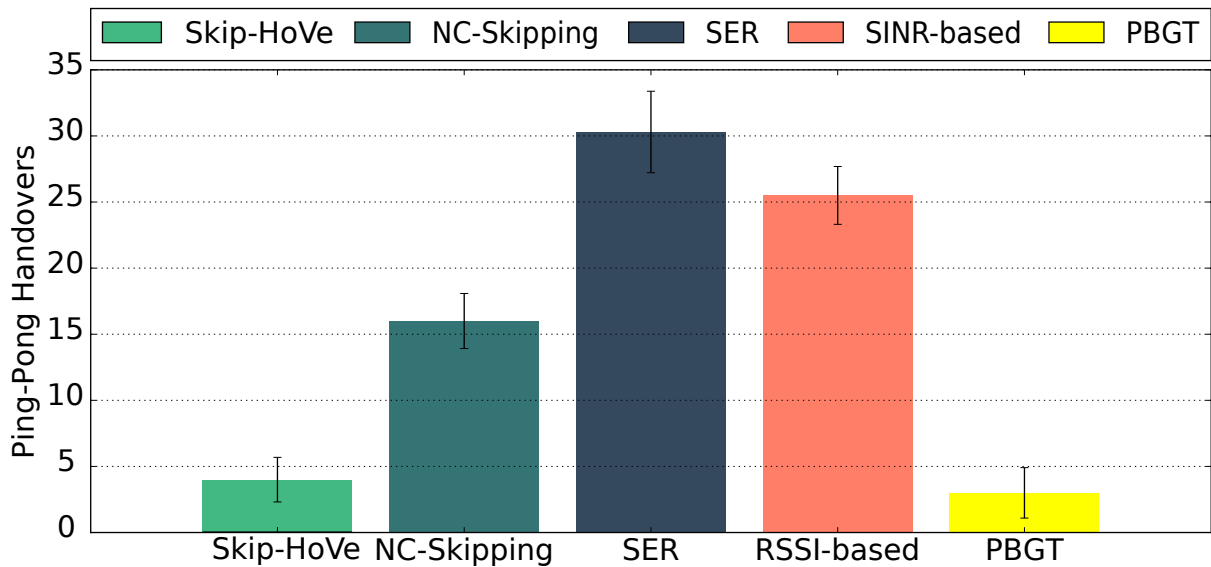


Figura 21 – Taxa de *handover* com efeito ping-pong por diferentes algoritmos de *handover*.

Analisando os resultados do algoritmo Skip-HoVe, das 38 conexões possíveis, o algoritmo realizou apenas 8 *handovers*, enquanto eram avaliadas para manter uma alta QoE. O NC-Skipping, efetua quase a mesma quantidade de *handover* que o Skip-HoVe, porém ele falha em manter QoE e QoS aceitáveis. No caso do SER, o algoritmo é muito sensível às flutuações na QoE, executando, portanto, um grande número de desconexões e *handover* com efeito de *ping-pong*, e ainda, uma alta taxa de perda de quadros I. O algoritmo SINR, por outro lado, é muito sensível às flutuações aleatórias de sinal. Acerca de tal algoritmo, quando um nó está em áreas de cobertura sobrepostas, os *handovers* são muito frequentes. Já o algoritmo PBGT é menos sensível às flutuações e geralmente prefere macrocélulas. No que diz respeito a esse algoritmo, as duas macrocélulas que o veículo atravessou acionaram uma conexão. Portanto, o algoritmo Skip-HoVe evita *handovers* desnecessárias e oferece uma experiência perfeita para os usuários finais.

Tabela 5 – Resumo de resultados para um veículo no cenário.

#	Algoritmo	MOS	Número de Handovers	Ping-Pong Handovers	Taxa de perda do Quadro I	Pequenas Células	Macro Células
01	Skip-HoVe	8	8	0	5%	38	2
02	NC-Skipping	1	10	3	66%	44	1
03	SER	1	36	21	64%	43	1
04	SINR-based	2	25	13	67%	43	2
05	PBGT	2	2	0	67%	49	2

5.3 Conclusão do Capítulo

Este Capítulo discutiu a metodologia de avaliação, na qual foi implementado o algoritmos Skip-HoVe na arquitetura do simulador NS-3, junto com a versão implementada na pilha de

protocolos LTE para comunicação V2V eV2I. Foi discutida a descrição do cenário de VANET ultradensa representado pelos dados coletados pela fonte *trace* na área da Baía de São Francisco, nos EUA, devido à sua relação direta com o mundo real. Um vídeo com níveis de complexidade moderada (sequência de vídeo *Highway*), em termos de movimento e complexidade espacial, foi aplicado nas simulação. Enfim, nas simulações foram utilizados parâmetros e métricas para avaliar o desempenho de diferentes algoritmos de *handover* destinados à distribuição de vídeo em um cenário de VANET ultradensa.

6 CONSIDERAÇÕES FINAIS

Este capítulo se baseia na conclusão geral desta Tese e discute as conduções para futuras pesquisas. O objetivo é destacar as principais contribuições e apontar inúmeras direções possíveis para prosseguir com a pesquisa. Será exposto primeiro as conclusões da tese na Seção 6.1. Em seguida, na Seção 6.2, será exibido as diretrizes futuras deste trabalho. Finalmente, na Seção 6.3, será apresentado a publicação relacionada e a complementar à esta Tese.

6.1 Conclusões Gerais

Nesta pesquisa foi realizada uma abordagem geral sobre um novo algoritmo baseado em *handover*, o qual possui vários critérios para distribuição de vídeo em cenários de VANET ultradensas, chamado Skip-HoVe. Tal algoritmo fornece decisões de *handover* ininterruptas, acoplando uma técnica de decisão de entrega com vários critérios para melhorar a QoE das transmissões de vídeo e reduzir as taxas de efeito *ping-pong*. Isto posto, o algoritmo Skip-HoVe, também, considera o modelo ARIMA para previsão de mobilidade do veículo, a PDR como critério de QoS, uma estimativa híbrida de QoE como parâmetro de QoE e o RSRQ como parâmetro de rádio.

Logo, acerca da decisão de entrega, o algoritmo Skip-HoVe calcula o nível de qualidade de cada célula para selecionar uma que seja a mais apropriada para realizar uma conexão com o veículo, considerando que o AHP atribui diferentes graus de importância no que diz respeito a cada critério. Sendo assim, por meio dessas abordagens, o algoritmo Skip-HoVe evita *handover* desnecessários e oferece uma experiência perfeita para os usuários finais. A análise de avaliação de desempenho apresentada revelou que o algoritmo Skip-HoVe melhorou a entrega de vídeo em até 14% na métrica SSIM, comparado com os algoritmos NC-Skipping, SER, SINR-based e PBGT, e os resultados do MOS mostraram uma avaliação subjetiva até 30% superior, enquanto mantinha a taxa de efeito *ping-pong* menor que 2%.

Outrossim, com a finalidade de garantir a disseminação de vídeo com QoE satisfatória em cenários de VANETs ultradensas, os resultados da simulação mostraram a superioridade e eficiência do algoritmo Skip-HoVe, comparado com os demais algoritmos analisados. Além disso, os resultados obtidos através de experimentos em cenários de VANETs demonstram que o algoritmo de *handover* Skip-HoVe é capaz de oferecer uma solução escalável e leve ao promover a disseminação inteligente de conteúdo multimídia em tempo real através do uso dos componentes incorporados a ele, tais como a predição de mobilidade, a QoS, a QoE e os que resultem em decisões de *handover* entre células com maior confiabilidade e menor frequência, que é o principal desafio em redes veiculares ultradensas, respondendo, assim, à questão principal da pesquisa.

6.2 Trabalhos Futuros

Com relação aos trabalhos futuros, serão realizadas novas avaliações com objetivo de estender o Skip-HoVe nas seguintes direções:

1. Ampliar a técnica de previsão de mobilidade e integrar o algoritmo de *handover* para todos os usuários da rede; além de, analisar e correlacionar os padrões de mobilidade de vários usuários, a fim de prever o congestionamento da área e executar o descarregamento eficiente de células e serviços de borda.
2. Avaliar o algoritmo Skip-HoVe com outros formatos de vídeo (HD, FHD, FUHD) aplicados em redes móveis, tanto para vídeos sob demanda quanto para vídeos em tempo real.
3. Usar outros métodos aplicado à inteligência artificial (por exemplo, Lógica Fuzzy (difusa ou nebulosa), Computação Evolutiva, Redes Neurais Artificiais, Teoria do Aprendizado, Métodos Estatísticos e Algoritmos Genéticos (evolutivos).
4. Usar AHP agregado a outros parâmetros como, por exemplo, tipo de veículo, histórico de dirigibilidade, velocidade e trajeto do veículo, carga de rede e intensidade do sinal das BSs.
5. Um aspecto interessante a ser avaliado é o paralelismo do algoritmo Skip-Hove, aumentando o seu desempenho em cenários de mobilidade com uma densidade maior de veículos.
6. Realizar um estudo da variação do tamanho do GoP, da paquetização e do nível de complexidade em diferentes vídeos.
7. Por fim, projetar soluções integradas nas quais os aplicativos possam se beneficiar do algoritmo Skip-HoVe e avaliar seu melhor desempenho.

6.3 Trabalhos Acadêmicos Publicados

O principal resultado obtido nesta Tese de Doutorado foi publicado no evento da *Computer Networks* que é uma revista internacional com cobertura completa de todos os tópicos de interesse para os envolvidos na área de Redes de Computadores.

- A. Costa, L. Pacheco, D. Rosário, L. Villas, A. Loureiro, S. Sargento e E. Cerqueira, "Skipping-based Handover Algorithm for Video Distribution Over Ultra-Dense VANET. *Computer Networks*¹. Volume 176, 20 July 2020, DOI: 101016.

¹ <https://doi.org/10.1016/j.comnet.2020.107252>

Outro resultado complementar a esta Tese de Doutorado, foi a publicação internacional no *International Journal of Distributed Sensor Networks* (IJDSN) que é um periódico indexado pela *Journal Citation Reports* (JCR) e revisado por grupos de pesquisadores que se concentram na pesquisa aplicada e na computação de alto desempenho.

- P. Bezerra, A. Costa, A. Melo, H. Santos, D. Rosário, E. Cerqueira, "A collaborative routing protocol for video streaming with fog computing in vehicular ad hoc networks."²Em *International Journal of Distributed Sensor Networks* [IJDSN]. 2019. Volume: 15 issue: 3. DOI: 101177.

² <https://doi.org/10.1177/1550147719832839>

REFERÊNCIAS

- 3GPP. Evolved universal terrestrial radio access (e-utra); radio resource control (rrc); protocol specification. *3GPP TS 36.331 V9.4.0 (2010-09), Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification (Release 9)*, 2011.
- ACCESS, E. U. T. R. **Study on Small Cell Enhancements for E-UTRA and E-UTRAN—Higher layer aspects (Release 12)**,”. [S.l.]: May, 2013.
- ALJERI, N.; BOUKERCHE, A. Movement prediction models for vehicular networks: an empirical analysis. **Wireless Networks**, Springer, v. 25, n. 4, p. 1505–1518, 2019.
- ALJERI, N.; BOUKERCHE, A. A two-tier machine learning-based handover management scheme for intelligent vehicular networks. **Ad Hoc Networks**, Elsevier, v. 94, p. 101930, 2019.
- ARGYRIOU, A. et al. Video delivery in dense 5g cellular networks. **IEEE Network**, v. 31, p. 28–34, 2017.
- ARSHAD, R. et al. Handover management in 5g and beyond: A topology aware skipping approach. **IEEE Access**, IEEE, v. 4, p. 9073–9081, 2016.
- ARSHAD, R. et al. Velocity-aware handover management in two-tier cellular networks. **IEEE Transactions on Wireless Communications**, IEEE, v. 16, n. 3, p. 1851–1867, 2017.
- BACCELLI, F.; BŁASZCZYSZYN, B. On a coverage process ranging from the boolean model to the poisson–voronoi tessellation with applications to wireless communications. **Advances in Applied Probability**, Cambridge University Press, v. 33, n. 2, p. 293–323, 2001.
- BEYRANVAND, H. et al. Toward 5g: Fiwi enhanced lte-a hetnets with reliable low-latency fiber backhaul sharing and wifi offloading. **IEEE/ACM Transactions on Networking**, IEEE, v. 25, n. 2, p. 690–707, 2017.
- BRUNELLI, M. **Introduction to the analytic hierarchy process**. [S.l.]: Springer, 2014.
- BUI, N. et al. A Survey of Anticipatory Mobile Networking: Context-Based Classification, Prediction Methodologies, and Optimization Techniques. **IEEE Communications Surveys and Tutorials**, v. 19, n. 3, p. 1790–1821, 2017. ISSN 1553877X.
- CERQUEIRA, E. et al. Multimedia human-centric networking: Concepts, technologies and trends. SBC, 2014.
- CHEN, J. et al. Qoe-aware intelligent vertical handoff scheme over heterogeneous wireless access networks. **IEEE Access**, IEEE, v. 6, p. 38285–38293, 2018.
- CHEN, K.-T.; TU, C.-C.; XIAO, W.-C. Oneclick - a framework for measuring network quality of experience. p. 702–710, 2009.
- CHEN, Q.; JIN, Y.; YANG, T. A supervised no-reference qoe assessment model on iptv services. In: IEEE. **2016 4th International Conference on Cloud Computing and Intelligence Systems (CCIS)**. [S.l.], 2016. p. 272–277.
- CHO, S.-G. et al. Image quality evaluation: Jpeg 2000 versus intra-only h. 264/avc high profile. **Facta universitatis-series: Electronics and Energetics**, v. 20, n. 1, p. 71–83, 2007.

- COSTA, A. et al. Skipping-based handover algorithm for video distribution over ultra-dense vanet. p. 1–9, 2020.
- COSTA, A. et al. Qoe-based packet dropper controllers for multimedia streaming in wimax networks. In: **Proceedings of the 6th Latin America Networking Conference**. [S.l.: s.n.], 2011. p. 12–19.
- DEMARCHOU, E.; PSOMAS, C.; KRIKIDIS, I. Mobility Management in Ultra-Dense Networks: Handover Skipping Techniques. **IEEE Access**, IEEE, v. 6, p. 11921–11930, 2018. ISSN 21693536.
- DÍAZ, C. et al. Xlr (pixel loss rate): a lightweight indicator to measure video qoe in ip networks. **IEEE Transactions on Network and Service Management**, IEEE, 2020.
- FELICE, M. D. et al. A distributed beaconless routing protocol for real-time video dissemination in multimedia vanets. **Computer communications**, Elsevier, v. 58, p. 40–52, 2015.
- GHAREEB, M.; VIHO, C. Hybrid qoe assessment is well-suited for multiple description coding video streaming in overlay networks. In: IEEE. **2010 8th Annual Communication Networks and Services Research Conference**. [S.l.], 2010. p. 327–333.
- GONG, F. et al. Cross-tier handover decision optimization with stochastic based analytical results for 5g heterogeneous ultra-dense networks. In: IEEE. **IEEE International Conference on Communications Workshops**. [S.l.], 2018. p. 1–6.
- GREENGRASS, J.; EVANS, J.; BEGEN, A. C. Not All Packets are Equal, Part 2: The Impact of Network Packet Loss on Video Quality. **IEEE Internet Computing**, IEEE, v. 13, n. 2, p. 74–82, 2009.
- GUAN, X. et al. Mobility aware partition of mec regions in wireless metropolitan area networks. In: IEEE. **IEEE Conference on Computer Communications Workshops (INFOCOM Workshop)**. [S.l.], 2018. p. 1–2.
- HUSIĆ, J. B.; BARAKOVIĆ, S.; OSMANOVIĆ, I. Subjective and objective qoe measurement for h. 265/hevc video streaming over lte. In: SPRINGER. **International Symposium on Innovative and Interdisciplinary Applications of Advanced Technologies**. [S.l.], 2019. p. 428–441.
- ITU, I.-T. R. G. . Reference guide to quality of experience assessment methodologies., **ITU-T**, 2010.
- ITU-T, R.; RECOMMEND, I. P. 800. **Methods for subjective determination of transmission quality**, 1996.
- JI, H. et al. Ultra-reliable and low-latency communications in 5g downlink: Physical layer aspects. **IEEE Wireless Communications**, IEEE, v. 25, n. 3, p. 124–130, 2018.
- KOTEVSKI, Z.; MITREVSKI, P. Performance assessment of metrics for video quality estimation. In: **Proc. Int. Sci. Conf. Inf. Commun. Energy Syst. Technol.(ICEST)**. [S.l.: s.n.], 2010. v. 1, p. 693–696.
- KURJENNIEMI, J.; HENTTONEN, T.; KAIKKONEN, J. Suitability of rsrq measurement for quality based inter-frequency handover in lte. In: IEEE. **2008 IEEE International Symposium on Wireless Communication Systems**. [S.l.], 2008. p. 703–707.

- LARRANAGA C, A. et al. The impact of compression in breast malignant tumor images and an ad hoc compressor proposal. **Recent Advances in Information Science**, 2016.
- LEE, Y. et al. Effects of time-to-trigger parameter on handover performance in son-based lte systems. In: IEEE. **2010 16th Asia-Pacific Conference on Communications (APCC)**. [S.l.], 2010. p. 492–496.
- LIOTOU, E. et al. Quality of experience management in mobile cellular networks: key issues and design challenges. **IEEE Communications Magazine**, IEEE, v. 53, n. 7, p. 145–153, 2015.
- LIU, Q. et al. A fuzzy-clustering based approach for madm handover in 5g ultra-dense networks. **Wireless Networks**, Springer, p. 1–14, 2019.
- LUO, G. et al. Cooperative vehicular content distribution in edge computing assisted 5G-VANET. **China Communications**, China Institute of Communications, v. 15, n. 7, p. 1–17, 2018. ISSN 16735447.
- MEDEIROS, I. et al. Handover ciente de qualidade de experiência e qualidade de serviço para transmissão de vídeo em redes heterogêneas. In: SBC. **Anais do XXIII Workshop de Gerência e Operação de Redes e Serviços**. [S.l.], 2018.
- MEDEIROS, I. et al. Quality of experience and quality of service-aware handover for video transmission in heterogeneous networks. **International Journal of Network Management**, Wiley Online Library, p. e2064, 2018.
- MOHAMED, A. et al. Mobility prediction for handover management in cellular networks with control/data separation. In: IEEE. **2015 IEEE International Conference on Communications (ICC)**. [S.l.], 2015. p. 3939–3944.
- MOHAMMADNIA, F. et al. Mobile small cells for adaptive ran densification: Preliminary throughput results. In: IEEE. **2019 IEEE Wireless Communications and Networking Conference (WCNC)**. [S.l.], 2019. p. 1–7.
- MONTGOMERY, C. et al. Xiph. org video test media (derf’s collection), the xiph open source community, <https://media.xiph.org/video/derf>, 1994. fev. 2020.
- MURUDKAR, C. V.; GITLIN, R. D. Qoe-driven anomaly detection in self-organizing mobile networks using machine learning. In: IEEE. **2019 Wireless Telecommunications Symposium (WTS)**. [S.l.], 2019. p. 1–5.
- NAGIN, K. et al. Estimating client qoe from measured network qos. In: **Proceedings of the 12th ACM International Conference on Systems and Storage**. [S.l.: s.n.], 2019. p. 188–188.
- NDASHIMYE, E. et al. Vehicle-to-infrastructure communication over multi-tier heterogeneous networks: A survey. **Computer networks**, Elsevier, v. 112, p. 144–166, 2017.
- NIGHTINGALE, J. et al. 5g-qoe: Qoe modelling for ultra-hd video streaming in 5g networks. **IEEE Transactions on Broadcasting**, IEEE, v. 64, n. 2, p. 621–634, 2018.
- NOIRIE, L. et al. Semantic networking: Flow-based, traffic-aware, and self-managed networking. **Bell Labs Technical Journal**, Wiley Online Library, v. 14, n. 2, p. 23–38, 2009.
- PACHECO, L. et al. A handover algorithm for video sharing over vehicular networks. In: IEEE. **anais do 9th Latin-American Symposium on Dependable Computing (LADC)**. [S.l.], 2019. p. 1–9.

- PIORKOWSKI, M.; SARAFIJANOVIC-DJUKIC, N.; GROSSGLAUSER, M. **CRAWDAD dataset epfl/mobility (v. 2009-02-24)**. 2009. Downloaded from <<https://crawdad.org/epfl/mobility/20090224>>.
- QUADROS, C. et al. A qoe-aware reinforcement approach to disseminate warning videos on lte-vanets. In: SBC. **Anais do XXXV Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos**. [S.l.], 2017.
- RECOMMENDATION, G. 1011. reference guide to quality of experience assessment methodologies. 2013.
- ROSÁRIO, D. et al. A qoe handover architecture for converged heterogeneous wireless networks. **Wireless networks**, Springer, v. 19, n. 8, p. 2005–2020, 2013.
- ROSÁRIO, D. et al. Trends in human-centric multimedia networking scenarios. In: IEEE. **2016 Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)**. [S.l.], 2016. p. 1–5.
- RUBIO, L.; REIG, J.; CARDONA, N. Evaluation of nakagami fading behaviour based on measurements in urban scenarios. **AEU-International Journal of Electronics and Communications**, Elsevier, v. 61, n. 2, p. 135–138, 2007.
- SAATY, T. L. Analytic hierarchy process. **Encyclopedia of Biostatistics**, Wiley Online Library, v. 1, 2005.
- SAATY, T. L. Decision making with the analytic hierarchy process. **International journal of services sciences**, v. 1, n. 1, p. 83–98, 2008.
- SARGENTO, S. et al. Method and apparatus for multi-network communication in vehicular networks. **US Patent US9439121 B2**, 2016.
- SHAH, S. A. A. et al. 5G for Vehicular Communications. **IEEE Communications Magazine**, v. 56, n. 1, p. 111–117, 2018. ISSN 01636804.
- SILVA, K. D. C.; BECVAR, Z.; FRANCES, C. R. L. Adaptive hysteresis margin based on fuzzy logic for handover in mobile networks with dense small cells. **IEEE Access**, IEEE, v. 6, p. 17178–17189, 2018.
- SUNG, M. et al. Realistic sonar image simulation using generative adversarial network. **IFAC-PapersOnLine**, Elsevier, v. 52, n. 21, p. 291–296, 2019.
- TARTARINI, L. et al. Software-defined handover decision engine for heterogeneous cloud radio access networks. **Computer Communications**, v. 115, p. 21 – 34, 2018. ISSN 0140-3664.
- THAKUR, P.; GANPATI, A. Survey on handover techniques in vanets. 2019.
- UNION, I. Itu-t recommendation p. 800.1: Mean opinion score (mos) terminology. **International Telecommunication Union, Tech. Rep**, 2006.
- WANG, T.; WANG, P.; LIU, B. Research on cross-layer routing protocol in vanet under urban environment. **Computer Engineering**, n. 11, p. 11, 2017.
- XU, X. et al. Delay-Oriented Cross-Tier Handover Optimization in Ultra-Dense Heterogeneous Networks. **IEEE Access**, IEEE, v. 7, p. 21769–21776, 2019. ISSN 21693536.

YILMAZ, G. N. Depth perception prediction of 3d video qoe for future internet services. In: IEEE. **2018 International Conference on Information Networking (ICOIN)**. [S.l.], 2018. p. 146–149.

ZHANG, D. et al. Aht: Application-based handover triggering for saving energy in cellular networks. In: IEEE. **2018 15th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)**. [S.l.], 2018. p. 1–9.

ZHENG, K. et al. Quality-of-experience assessment and its application to video services in lte networks. **IEEE Wireless Communications**, IEEE, v. 22, n. 1, p. 70–78, 2015.

Apêndices A - Código Desenvolvido

Allan Douglas Bento da Costa

INFORMAÇÕES

O algoritmo Skip-HoVe foi desenvolvido na linguagem GNU Bash é um interpretador de comandos. o Bash é uma evolução retro-compatível muito mais interativa do Bourne Shell (sh). Os shells Bourne permitem a execução de sequências de comandos inseridos diretamente na linha de comandos ("prompt") ou ainda lidos de arquivos de texto conhecidos como shell scripts. O algoritmo Skip HoVe foi implementado na arquitetura do *Network simulator* (NS), versão 3.29¹ e está disponível para download no Github². Cabe registrar que na respectiva versão está implementada a pilha de protocolos LTE para comunicação V2V e V2I. Um cenário de VANET ultradensa foi considerada na simulação, juntamente com o pacote 3GPP LTE release 13, observando uma área de 2km x 2km com 7 macrocélulas que cobrem todo o cenário até certo ponto e 100 células pequenas distribuídas por toda área. Os principais parâmetros de simulação. Abaixo, segue o *script* de execução das simulações recorrentes durante a gerações de resultados.

1 Código de Programação em GNU Bash

```
1 #!/bin/bash
2 #default values
3 n=1
4 video="highway"
5 ue_n=60
6 enb_n=100
7
8 dir=${PWD##*/}
9 if [ $dir == "Skip-HoVe" ] ||
10    [ $dir == "NC-Skipping" ] ||
11    [ $dir == "SER" ] ||
12    [ $dir == "SINR-based" ] ||
13    [ $dir == "PGBT" ]
14 then
15     alg=$dir
16 else
17     alg="Skip-HoVe"
18 fi
19
20 # check if arguments were passed
```

¹<http://www.nsnam.org/>

²<https://github.com/allandoug/>


```

21 if [ $# -gt 0 ]
22 then
23     # read command line arguments
24     for i in "$@"
25     do
26         case $i in
27             -n=*|--simul-number=*)
28                 n="${i#*=}"
29                 shift
30                 ;;
31             -alg=*|--algorithm=*)
32                 alg="${i#*=}"
33                 shift
34                 ;;
35             -v=*|--video=*)
36                 video="${i#*=}"
37                 shift
38                 ;;
39             -uen=*|--ue-number=*)
40                 ue_n="${i#*=}"
41                 shift
42                 ;;
43             -enbn=*|--enb-number=*)
44                 enb_n="${i#*=}"
45                 shift
46                 ;;
47             *)
48                 ;;
49         esac
50     done
51
52 else
53     # if no arguments, read one by one
54     echo -n "how many times do you wish to run? "
55     read n
56     echo -n "whats the algorithm used? "
57     read alg
58     echo -n "Video to be used: "
59     read video
60 fi
61
62 # translate video to id
63 case $video in
64     "highway")
65         video_id=1;;
66     "container")
67         video_id=2;;
68     "highway600")
69         video_id=3;;
70     "akiyo")
71         video_id=4;;
72     "masha")
73         video_id=7;;
74     "babyshark")

```

```

75     video_id=8;;
76     "despacito")
77     video_id=9;;
78 *)
79     exit 1
80 esac
81
82 # change id in file
83 sed -i "s/\(define video \)[0-9]/\1$video_id/g" scratch/
    v2x_3gpp_small.cc
84 sed -i "s/\(node_ue = \)[0-9]\{,3\}/\1$ue_n/g" scratch/
    v2x_3gpp_small.cc
85 sed -i "s/\(low_power = \)[0-9]\{,3\}/\1$enb_n/g" scratch/
    v2x_3gpp_small.cc
86
87
88 [ -d ${alg}_${enb_n} ] || mkdir ${alg}_${enb_n}
89
90 for i in $(seq 1 $n)
91 do
92     echo "running for seed value ${i}."
93     echo "outputting to file: log_${alg}_${i}"
94
95     echo "Configuring for:" | tee log_${alg}_${i}
96     echo "  n=${n}" | tee -a log_${alg}_${i}
97     echo "  video=${video}" | tee -a log_${alg}_${i}
98     echo "  ue_n=${ue_n}" | tee -a log_${alg}_${i}
99     echo "  enb_n=${enb_n}" | tee -a log_${alg}_${i}
100    echo "  alg=${alg}" | tee -a log_${alg}_${i}
101    echo
102
103    # run simulation
104    ./waf configure >> log_${alg}_${i} 2>&1
105    echo "configuration left with status ${?}"
106    echo
107
108    time ./waf --run "scratch/v2x_3gpp_small --handoverAlg=
        $alg --seedValue=${i}" >> log_${alg}_${i} 2>&1
109    echo "simulation left with status ${?}"
110    mkdir -p ${alg}_${enb_n}/simul${i}
111    mv v2x_temp log_${alg}_${i} ${alg}_${enb_n}/simul${i}
112 done

```

Listing 1: Código Fonte em GNU Bash

Apêndices B - Código Desenvolvido

Allan Douglas Bento da Costa

INFORMAÇÕES

O algoritmo Skip-HoVe foi desenvolvido na linguagem de programação C, seu uso é de propósito geral, estruturada, imperativa, procedural, padronizada pela ISO. O algoritmo Skip HoVe foi implementado na arquitetura do *Network simulator* (NS), versão 3.29¹ e está disponível para download no Github². Cabe registrar que na respectiva versão está implementada a pilha de protocolos LTE para comunicação V2V e V2I. Um cenário de VANET ultradensa foi considerada na simulação, juntamente com o pacote 3GPP LTE release 13, observando uma área de 2km x 2km com 7 macrocélulas que cobrem todo o cenário até certo ponto e 100 células pequenas distribuídas por toda área. Os principais parâmetros de simulação. Abaixo, segue o principal código desenvolvido (existe outros) nas simulações recorrentes durante a gerações de resultados.

1 Código do Algoritmo Skip-Hove

```
1 #include "ns3/core-module.h"
2 #include "ns3/network-module.h"
3 #include "ns3/internet-module.h"
4 #include "ns3/mobility-module.h"
5 #include "ns3/applications-module.h"
6 #include "ns3/config-store-module.h"
7 #include "ns3/point-to-point-module.h"
8 #include "ns3/ipv4-global-routing-helper.h"
9 #include "ns3/ipv4-address.h"
10 #include "ns3/mobility-model.h"
11 // NetAnim & Evalvid
12 #include "ns3/netanim-module.h"
13 #include "ns3/evalvid-client-server-helper.h"
14 // Pacotes LTE
15 #include "ns3/point-to-point-helper.h"
16 #include "ns3/lte-helper.h"
17 #include "ns3/epc-helper.h"
18 #include "ns3/lte-module.h"
19 #include "ns3/string.h"
20 #include "ns3/double.h"
21 #include <ns3/boolean.h>
22 #include <ns3/enum.h>
23 #include <iomanip>
```

¹<http://www.nsnam.org/>

²<https://github.com/allandoug/>

```
24 #include <string>
25 #include <fstream>
26 #include <iostream>
27 #include <vector>
28 #include <stdio.h>
29 #include <stdlib.h>
30 #include <memory>
31 #include <typeinfo>
32 #include <math.h>
33 #define PI 3.14159265
34 #define SIMULATION_TIME_FORMAT(s) Seconds(s)
35 #define video 1
36
37 #if video == 1
38 #define video_st "sourceTraces/st_highway_cif.st"
39 #define numberOfFrames 2000
40 #define numberOfPackets 2106
41 #define gop 30
42
43 #elif video == 2
44 #define video_st "sourceTraces/st_container_cif_h264_300_20.
    st"
45 #define numberOfFrames 300
46 #define numberOfPackets 614
47 #define gop 20
48
49 #elif video == 3
50 #define video_st "sourceTraces/st_highway_600_cif"
51 #define numberOfFrames 600
52 #define numberOfPackets 716
53 #define gop 28
54
55 #elif video == 4
56 #define video_st "sourceTraces/st_akiyo_cif_h264_300_18"
57 #define numberOfFrames 300
58 #define numberOfPackets 839
59 #define gop 18
60
61 #elif video == 5
62 #define video_st "sourceTraces/st_football_cif"
63 #define numberOfFrames 260
64 #define numberOfPackets 537
65 #define gop 28
66
67 #elif video == 6
68 #define video_st "sourceTraces/st_mobile_cif_h264_300_18"
69 #define numberOfFrames 300
70 #define numberOfPackets 605
71 #define gop 28
72
73 #endif
74
75 using namespace ns3;
76
```

```

77 // do not set this variables, for control only
78 double TxRate = 0;
79 unsigned int handNumber = 0;
80 double qosLastValue = 0;
81 double qoeLastValue = 0;
82 int evalvidId = 0;
83
84 // simulation options
85 bool verbose = true;
86 double simTime = 2; // simlation time
87 int transmissionStart = 7; // transmission time
88
89 // number of user devices
90 const int node_ue = 100;
91
92 // number os cells of each kind
93 const uint16_t enb_HPN = 2;
94 const uint16_t low_power = 100;
95 const uint16_t hot_spot = 0;
96
97 // tranmission power for cells
98 int hpnTxPower = 46;
99 int lpnTxPower = 23;
100 int hpTxPower = 15;
101 const int node_enb = enb_HPN + low_power + hot_spot;
102
103 // CBR setting
104 bool useCbr = false;
105 uint16_t n_cbr = useCbr ? enb_HPN + low_power : 0;
106
107 // topology type
108 bool randomCellAlloc = true; // random uniform
109 bool rowTopology = false; // aligned
110
111 // control matrix
112 int cell_ue[enb_HPN + low_power + hot_spot][node_ue];
113
114 // exp average parameter
115 unsigned int exp_mean_window = 3;
116
117 // variables for metrics computation
118 double qoeSum[enb_HPN + low_power + hot_spot];
119 double qosSum[enb_HPN + low_power + hot_spot];
120 int qosMetricsIterator[enb_HPN + low_power + hot_spot];
121 int qoeMetricsIterator[enb_HPN + low_power + hot_spot];
122
123 int framePct[numberOfFrames + 1];
124 std::string frameTypeGlobal[numberOfFrames];
125 int LastReceivedFrame[node_ue];
126 bool receivedFrames[node_ue][numberOfPackets];
127 bool receivedPackets[node_ue][numberOfPackets];
128 /*
-----
*/

```

```

129
130 NS_LOG_COMPONENT_DEFINE("v2x_3gpp");
131
132 /*----- NOTIFICACOES DE HANDOVER
-----*/
133 void NotifyConnectionEstablishedUe(std::string context,
134                                     uint64_t imsi,
135                                     uint16_t cellid,
136                                     uint16_t rnti)
137 {
138     NS_LOG_DEBUG(Simulator::Now().GetSeconds()
139                 << " " << context << " UE IMSI " << imsi
140                 << ": connected to CellId " << cellid << "
141                 with RNTI " << rnti);
142
143     std::stringstream temp_cell_dir;
144     std::stringstream ueId;
145     temp_cell_dir << "./v2x_temp/" << cellid;
146     ueId << temp_cell_dir.str() << "/" << rnti;
147     if (mkdir(temp_cell_dir.str().c_str(), S_IRWXU | S_IRWXG
148             | S_IROTH | S_IXOTH) != 0)
149     {
150     }
151     std::ofstream outfile(ueId.str().c_str());
152     outfile << imsi << std::endl;
153     outfile.close();
154
155     cell_ue[cellid - 1][imsi - 1] = rnti;
156 }
157
158 void NotifyHandoverStartUe(std::string context,
159                             uint64_t imsi,
160                             uint16_t cellid,
161                             uint16_t rnti,
162                             uint16_t targetCellId)
163 {
164     NS_LOG_DEBUG(Simulator::Now().GetSeconds()
165                 << " " << context << " UE IMSI " << imsi
166                 << ": previously connected to CellId " <<
167                 cellid << " with RNTI "
168                 << rnti << ", doing handover to CellId " <<
169                 targetCellId);
170
171     cell_ue[cellid - 1][imsi - 1] = 0;
172
173     std::stringstream ueId;
174     ueId << "./v2x_temp/" << cellid << "/" << rnti;
175     remove(ueId.str().c_str());
176
177     ++handNumber;
178 }
179
180 void NotifyHandoverEndOkUe(std::string context,
181                             uint64_t imsi,

```

```

178         uint16_t cellid,
179         uint16_t rnti)
180 {
181     NS_LOG_DEBUG(Simulator::Now().GetSeconds()
182                 << " " << context << " UE IMSI " << imsi
183                 << ": successful handover to CellId " <<
184                 cellid << " with RNTI "
185                 << rnti);
186
187     std::stringstream target_cell_dir;
188     std::stringstream newUeId;
189     target_cell_dir << "./v2x_temp/" << cellid;
190     newUeId << target_cell_dir.str() << "/" << rnti;
191     if (mkdir(target_cell_dir.str().c_str(), S_IRWXU |
192             S_IRWXG | S_IROTH | S_IXOTH) != 0)
193     {
194     }
195     std::ofstream outfile(newUeId.str().c_str());
196     outfile << imsi << std::endl;
197     outfile.close();
198
199     cell_ue[cellid - 1][imsi - 1] = rnti;
200 }
201
202 void NotifyConnectionEstablishedEnb(std::string context,
203                                     uint64_t imsi,
204                                     uint16_t cellid,
205                                     uint16_t rnti)
206 {
207     NS_LOG_DEBUG(Simulator::Now().GetSeconds()
208                 << " " << context << " eNB CellId " <<
209                 cellid
210                 << ": successful connection of UE with IMSI
211                 " << imsi << " RNTI "
212                 << rnti);
213 }
214
215 void NotifyHandoverStartEnb(std::string context,
216                             uint64_t imsi,
217                             uint16_t cellid,
218                             uint16_t rnti,
219                             uint16_t targetCellId)
220 {
221     NS_LOG_DEBUG(Simulator::Now().GetSeconds()
222                 << " " << context << " eNB CellId " <<
223                 cellid
224                 << ": start handover of UE with IMSI " <<
225                 imsi << " RNTI "
226                 << rnti << " to CellId " << targetCellId);
227 }
228
229 void NotifyHandoverEndOkEnb(std::string context,
230                              uint64_t imsi,
231                              uint16_t cellid,

```

```

226         uint16_t rnti)
227     {
228         NS_LOG_DEBUG(Simulator::Now().GetSeconds()
229             << " " << context << " eNB CellId " <<
230                 cellid
231                 << ": completed handover of UE with IMSI "
232                 << imsi << " RNTI "
233                 << rnti);
234     }
235
236 void ArrayPositionAllocator(Ptr<ListPositionAllocator>
237     HpnPosition)
238 {
239     int x, y;
240     int distance = 1000;
241     std::ofstream outfile("v2x_temp/cellsList", std::ios::
242         out | std::ios::trunc);
243
244     if (randomCellAlloc)
245     {
246         for (int i = 0; i < enb_HPN + low_power + hot_spot;
247             ++i)
248         {
249             x = rand() % 100;
250             y = rand() % 2000;
251             HpnPosition->Add(Vector(x, y, 15));
252             outfile << i + 1 << " " << x << " " << y << std
253                 ::endl;
254         }
255         outfile.close();
256         return;
257     }
258
259     else if (rowTopology)
260     {
261         int x_start = 700;
262         int y_start = 500;
263         for (int i = 0; i < enb_HPN + low_power + hot_spot;
264             ++i)
265             HpnPosition->Add(Vector(x_start + distance * i,
266                 y_start, 25));
267         return;
268     }
269
270     else
271     {
272         int x_start = 1000;
273         int y_start = 1000;
274
275         HpnPosition->Add(Vector(x_start, y_start, 25));
276
277         for (double i = 0; i < 2 * PI; i += PI / 3)
278         {
279             HpnPosition->Add(Vector(x_start + distance * cos

```



```

272         (i), y_start + distance * sin(i), 25));
273     }
274     for (double i = 0; i < 2 * PI; i += PI / 3)
275     {
276         HpnPosition->Add(Vector(x_start + distance * cos
277             (i) + rand() % 100 + 10, y_start + distance *
278             sin(i) + rand() % 100 + 10, 10));
279         HpnPosition->Add(Vector(x_start + distance * cos
280             (i) + rand() % 100 + 10, y_start + distance *
281             sin(i) - rand() % 100 + 10, 10));
282         HpnPosition->Add(Vector(x_start + distance * cos
283             (i) + rand() % 100 - 10, y_start + distance *
284             sin(i) + rand() % 100 + 10, 10));
285         HpnPosition->Add(Vector(x_start + distance * cos
286             (i) + rand() % 100 - 10, y_start + distance *
287             sin(i) - rand() % 100 + 10, 10));
288     }
289 }
290
291 void VideoTraceParse(std::string m_videoTraceFileName)
292 {
293     std::ifstream VideoTraceFile(m_videoTraceFileName.c_str
294         (), ios::in);
295     if (VideoTraceFile.fail())
296     {
297         NS_FATAL_ERROR(">> EvalvidServer: Error while
298             opening video trace file: "
299             << m_videoTraceFileName.c_str());
300         return;
301     }
302
303     uint32_t frameId;
304     std::string frameType;
305     uint32_t frameSize;
306     uint16_t numOfUdpPackets;
307     double sendTime;
308     uint32_t packet = 0;
309
310     while (VideoTraceFile >> frameId >> frameType >>
311         frameSize >> numOfUdpPackets >> sendTime)
312     {
313         framePct[frameId - 1] = packet + 1;
314         frameTypeGlobal[frameId - 1] = frameType;
315         packet += numOfUdpPackets;
316     }
317 }
318
319 std::string exec(const char *cmd)
320 {
321     std::array<char, 128> buffer;
322     std::string result;
323     std::shared_ptr<FILE> pipe(popen(cmd, "r"), pclose);

```

```

314     if (!pipe)
315         throw std::runtime_error("popen() failed!");
316     while (!feof(pipe.get()))
317     {
318         if (fgets(buffer.data(), 128, pipe.get()) != nullptr
319             )
320             result += buffer.data();
321     }
322     return result;
323 }
324 /*----- CRIA O ARQUIVO COM QOS'S -----*/
325 void WriteMetrics()
326 {
327     NS_LOG_DEBUG(Simulator::Now().GetSeconds() << " Segundos
328         ...");
329     NS_LOG_DEBUG("Realizados " << handNumber << " Handover")
330     ;
331     for (int i = 0; i < node_enb; ++i)
332     {
333         for (int u = 0; u < node_ue; ++u)
334         if (cell_ue[i][u])
335         {
336             std::stringstream rdTrace;
337             rdTrace << "v2x_temp/rd_a01_" << u;
338             std::ifstream rdFile(rdTrace.str());
339             if (!rdFile)
340             {
341                 NS_LOG_DEBUG("NO FILE TO BE READ" << '\n
342                     ');
343                 return;
344             }
345             double rdTime;
346             std::string id;
347             int rdPacketId;
348             string a;
349             int b;
350
351             int lastPacket = 0;
352             int counter;
353             int npackets;
354
355             while (rdFile >> rdTime >> id >> rdPacketId
356                 >> a >> b)
357             {
358                 receivedPackets[u][rdPacketId - 1] =
359                     true;
360                 lastPacket = rdPacketId;
361             }
362
363             /*-----QOS METRIC CALCULATION-----*/
364             int nReceived = 0;
365             for (int j = lastPacket - 60; j <=
366                 lastPacket; ++j)

```

```

361     {
362         if (receivedPackets[u][j])
363             ++nReceived;
364     }
365     if (lastPacket >= 60)
366     {
367         *NS_LOG_INFO("Taxa de recebimento, node
368             " << u << " : "
369             << ((float)nReceived - 1) / 60);*/
370         /NS_LOG_INFO ("Recebidos " << nReceived
371             << " pacotes.");
372         stringstream qosFilename;
373         double valorAtualQos = 0;
374         qosFilename << "./v2x_temp/qosTorre" <<
375             i + 1;
376         ifstream qosInFile(qosFilename.str());
377         while (qosInFile >> valorAtualQos)
378         {
379             }
380         ofstream qosOutFile(qosFilename.str(),
381             std::ofstream::out |
382             std::ofstream::
383             trunc);
384
385         //C LCULO DA M DIA EXPONENCIAL
386         qosOutFile << 2 * (((float)nReceived -
387             1) / 60 - valorAtualQos) / (
388             exp_mean_window + 1) + valorAtualQos;
389         NS_LOG_DEBUG("NODE " << u << " QOS
390             ESTIMADO " << 2 * (((float)nReceived
391             - 1) / 60 - valorAtualQos) / (
392             exp_mean_window + 1) + valorAtualQos)
393             ;
394
395         CALCULO DE QOS POR M DIA SIMPLES
396         qosSum[i] += ((float)nReceived - 1) /
397             60;
398         qosMetricsIterator[i]++;
399         qosOutFile << qosSum[i] /
400             qosMetricsIterator[i];
401         NS_LOG_DEBUG("NODE " << u << " QOS
402             ESTIMADO " << qosSum[i] /
403             qosMetricsIterator[i]);
404     }
405
406     /**-----*/
407     /*-----QOE METRIC CALCULATION-----*/
408     /*-----*/
409
410     framePct[numberOfFrames] = numberOfPackets +
411         1;
412     for (int j = 0; j < numberOfFrames; ++j)
413     {
414         npackets = framePct[j + 1] - framePct[j]

```

```

399         ];
400         counter = 0;
401         if (framePct[j] > lastPacket)
402         {
403             continue;
404         }
405         for (int k = framePct[j]; k < framePct[j]
406             + 1]; ++k)
407             if (receivedPackets[u][k - 1])
408             {
409                 ++counter;
410             }
411         if (npackets == counter)
412         {
413             receivedFrames[u][j] = true;
414             LastReceivedFrame[u] = j + 1;
415         }
416     }
417
418     int lastGop = 0;
419     int IReceived = 0;
420     int ITotal = 0;
421     double ILoss = 0;
422
423     int PReceived = 0;
424     int PTotal = 0;
425     double PLoss = 0;
426
427     int BReceived = 0;
428     int BTotal = 0;
429     double BLoss = 0;
430
431     for (int j = 0; j < LastReceivedFrame[u] /
432         20; j++)
433         ++lastGop;
434
435     if (lastGop != 0)
436     {
437         for (int j = 20 * (lastGop - 1); j <
438             lastGop * 20 - 1; ++j)
439         {
440             if (frameTypeGlobal[j].find("I") !=
441                 std::string::npos ||
442                 frameTypeGlobal[j].find("H") !=
443                 std::string::npos)
444                 ++ITotal;
445             else if (frameTypeGlobal[j].find("P"
446                 ) != std::string::npos)
447                 ++PTotal;
448             else if (frameTypeGlobal[j].find("B"
449                 ) != std::string::npos)
450                 ++BTotal;

```

```

444         if (receivedFrames[u][j])
445         {
446             if (frameTypeGlobal[j].find("I")
447                 != std::string::npos ||
448                 frameTypeGlobal[j].find("H")
449                 != std::string::npos)
450                 ++IReceived;
451             else if (frameTypeGlobal[j].find
452                 ("P") != std::string::npos)
453                 ++PReceived;
454             else if (frameTypeGlobal[j].find
455                 ("B") != std::string::npos)
456                 ++BReceived;
457         }
458     }
459     ILoss = ITotal != 0 ? ((double)ITotal -
460         (double)IReceived) * 100 / (double)
461         ITotal : 0;
462     PLoss = PTotal != 0 ? ((double)PTotal -
463         (double)PReceived) * 100 / (double)
464         PTotal : 0;
465     BLoss = BTotal != 0 ? ((double)BTotal -
466         (double)BReceived) * 100 / (double)
467         BTotal : 0;
468
469     std::stringstream cmd;
470     std::stringstream qoeFileName;
471     std::string qoeResult;
472
473     qoeFileName << "./v2x_temp/qoeTorre" <<
474         i + 1;
475     double valorAtualQoe = 0;
476     ifstream qoeInFile(qoeFileName.str());
477     while (qoeInFile >> valorAtualQoe)
478     {
479     }
480     // ESTE ARQUIVO CONT M O VALOR DA
481     // LTIMA M DIA CALCULADA
482     ofstream qoeOutFile(qoeFileName.str(),
483         std::ofstream::out |
484         std::ofstream::
485         trunc);
486
487     cmd << "python2.7 ia.py " << ILoss << "
488         " << PLoss << " " << BLoss
489         << " " << gop << " 2>/dev/null";
490
491     //C LCULO DA M DIA EXPONENCIAL E
492     // ESCRITA NO ARQUIVO
493     qoeOutFile << 2 * (stod(exec(cmd.str().
494         c_str())) - valorAtualQoe) / (
495         exp_mean_window + 1) + valorAtualQoe;
496     NS_LOG_INFO("NODE " << u << " QOE

```

```

479         ESTIMADO " << 2 * (stod(exec(cmd.str
480             (.c_str())) - valorAtualQoe) / (
481             exp_mean_window + 1) + valorAtualQoe)
482         ;
483         qoeOutFile.close();
484
485         C LCULO POR M DIA SIMPLES
486         qoeSum[i] += stod(exec(cmd.str().c_str
487             ());
488         qoeMetricsIterator[i]++;
489         qoeOutFile << qoeSum[i] /
490             qoeMetricsIterator[i];
491
492         NS_LOG_INFO("NODE " << u << " QOE
493             ESTIMADO " << qoeSum[i] /
494             qoeMetricsIterator[i]);
495     }
496 }
497
498 NS_LOG_INFO("\n\n\n\n");
499 return;
500 }
501
502 void requestStream(Ptr<Node> remoteHost, NodeContainer
503 ueNodes, Ipv4Address remoteHostAddr, double simTime,
504 double start)
505 {
506     for (uint32_t i = 0; i < ueNodes.GetN(); ++i)
507     {
508         string video_trans = video_st;
509         int startTime = rand() % 10 + transmissionStart;
510         NS_LOG_UNCOND("Node " << i << " requesting video at
511             " << startTime << "\n");
512         std::stringstream sdTrace;
513         std::stringstream rdTrace;
514         std::stringstream rdWindow;
515         sdTrace << "v2x_temp/sd_a01_" << evalvidId;
516         rdTrace << "v2x_temp/rd_a01_" << evalvidId;
517         evalvidId++;
518
519         double stop = simTime;
520         uint16_t m_port = 2000 * evalvidId + 2000; // Para
521             alcan ar o n ZERO quando i = 0
522
523         // Servidor de v deo
524         EvalvidServerHelper server(m_port);
525         server.SetAttribute("SenderTraceFilename",
526             StringValue(video_trans));
527         server.SetAttribute("SenderDumpFilename",
528             StringValue(sdTrace.str()));
529         server.SetAttribute("PacketPayload", UIntegerValue
530             (512));
531         ApplicationContainer apps = server.Install(
532             remoteHost);

```

```

517     apps.Start(Seconds(1));
518     apps.Stop(Seconds(stop));
519
520     // Clientes do v deo
521     EvalvidClientHelper client(remoteHostAddr, m_port);
522     client.SetAttribute("ReceiverDumpFilename",
523         StringValue(rdTrace.str()));
524     apps = client.Install(ueNodes.Get(i));
525     apps.Start(Seconds(startTime));
526     apps.Stop(Seconds(stop));
527
528     Ptr<Ipv4> ipv4 = ueNodes.Get(i)->GetObject<Ipv4>();
529 }
530
531 int main(int argc, char *argv[])
532 {
533
534     if (mkdir("./v2x_temp", S_IRWXU | S_IRWXG | S_IROTH |
535         S_IXOTH) != 0)
536     {
537         system("exec rm -rf v2x_temp/*");
538
539         for (int u = 0; u < node_ue; ++u)
540         {
541             for (int i = 0; i < numberOfPackets; ++i)
542                 receivedPackets[u][i] = false;
543
544             for (int i = 0; i < numberOfFrames; ++i)
545                 receivedFrames[u][i] = false;
546         }
547
548         int seedValue = 1;
549         std::string handoverAlg = "hove";
550         CommandLine cmm;
551
552         cmm.AddValue("seedValue", "random seed value.",
553             seedValue);
554         cmm.AddValue("handoverAlg", "Handover algorithm in use.",
555             handoverAlg);
556         cmm.AddValue("verbose", "lte stats verbose.", verbose);
557         cmm.Parse(argc, argv);
558         RngSeedManager::SetSeed(seedValue + 10000); //valor de
559             seed para gera o de n meros aleat rios
560         srand(seedValue);
561
562         double interPacketInterval = 1;
563         VideoTraceParse(video_st);
564
565         LogComponentEnable("v2x_3gpp", LOG_LEVEL_DEBUG);
566         LogComponentEnable("v2x_3gpp", LOG_LEVEL_INFO);
567         LogComponentEnable("EvalvidClient", LOG_LEVEL_INFO);
568         if (verbose)

```

```

566     {
567         LogComponentEnable("LteEnbRrc", LOG_LEVEL_ALL);
568         LogComponentEnable("LteUeRrc", LOG_LEVEL_ALL);
569         LogComponentEnable("LteUePhy", LOG_LEVEL_ALL);
570         LogComponentEnable("LteEnbPhy", LOG_LEVEL_ALL);
571         LogComponentEnable("HoveHandoverAlgorithm",
572             LOG_LEVEL_INFO);
573         LogComponentEnable("HoveHandoverAlgorithm",
574             LOG_LEVEL_DEBUG);
575     }
576
577     uint16_t node_remote = 1; // HOST_REMOTE
578     for (double t = transmissionStart; t < simTime; t += 1)
579         Simulator::Schedule(Seconds(t), &WriteMetrics);
580
581     ConfigStore inputConfig;
582     inputConfig.ConfigureDefaults();
583
584     /*---- LTE MODULE ----*/
585     Ptr<LteHelper> lteHelper = CreateObject<LteHelper>();
586     Ptr<PointToPointEpcHelper> epcHelper = CreateObject<
587         PointToPointEpcHelper>();
588     // EPC helper
589     Ptr<Node> pgw = epcHelper->GetPgwNode();
590     lteHelper->SetEpcHelper(epcHelper);
591
592     /****** CONFIGURA O LTE *****/
593
594     // bandwidth in resource blocks
595     Config::SetDefault("ns3::LteEnbNetDevice::DlBandwidth",
596         UintegerValue(25));
597     Config::SetDefault("ns3::LteEnbNetDevice::UlBandwidth",
598         UintegerValue(25));
599
600     // ctrl and data channel models
601     Config::SetDefault("ns3::LteSpectrumPhy::
602         CtrlErrorModelEnabled", BooleanValue(true));
603
604     // transmission mode (SISO [0], MIMO [1])
605     Config::SetDefault("ns3::LteEnbRrc::
606         DefaultTransmissionMode",
607         UintegerValue(1));
608
609     // path loss model
610     lteHelper->SetAttribute("PathlossModel",
611         StringValue("ns3::
612             NakagamiPropagationLossModel"
613         ));
614
615     // UL and DL frequencies
616     lteHelper->SetEnbDeviceAttribute("DlEarfcn",
617         UintegerValue(100));
618     lteHelper->SetEnbDeviceAttribute("UlEarfcn",
619         UintegerValue(18100));
620     lteHelper->SetSchedulerType("ns3::PssFfMacScheduler");
621     // the maximum number of UE selected by TD scheduler

```



```

609     lteHelper->SetSchedulerAttribute("nMux", UIntegerValue
        (1));
610     // PF scheduler type in PSS
611     lteHelper->SetSchedulerAttribute("PssFdSchedulerType",
        StringValue("CoItA"));
612
613     // RRC model
614     Config::SetDefault("ns3::LteHelper::UseIdealRrc",
        BooleanValue(true));
615
616     // Antenna parameters
617     lteHelper->SetEnbAntennaModelType("ns3::
        CosineAntennaModel");
618     lteHelper->SetEnbAntennaModelAttribute("Orientation",
        DoubleValue(0));
619     lteHelper->SetEnbAntennaModelAttribute("Beamwidth",
        DoubleValue(60));
620     lteHelper->SetEnbAntennaModelAttribute("MaxGain",
        DoubleValue(0.0));
621
622     // Handover algorithm selection
623     if (handoverAlg == "hove")
624     {
625         lteHelper->SetHandoverAlgorithmType("ns3::
        HoveHandoverAlgorithm");
626         lteHelper->SetHandoverAlgorithmAttribute("
        NumberOfUEs", UIntegerValue(node_ue));
627     }
628
629     else if (handoverAlg == "skip")
630     {
631         lteHelper->SetHandoverAlgorithmType("ns3::
        SkippingHandoverAlgorithm");
632         lteHelper->SetHandoverAlgorithmAttribute("
        NumberOfUEs", UIntegerValue(node_ue));
633     }
634
635     else if (handoverAlg == "ser")
636     {
637         lteHelper->SetHandoverAlgorithmType("ns3::
        SerHandoverAlgorithm");
638         lteHelper->SetHandoverAlgorithmAttribute("
        NumberOfUEs", UIntegerValue(node_ue));
639     }
640
641     else if (handoverAlg == "qoe")
642         lteHelper->SetHandoverAlgorithmType("ns3::
        QoeHandoverAlgorithm");
643
644     else if (handoverAlg == "noop")
645         lteHelper->SetHandoverAlgorithmType("ns3::
        NoOpHandoverAlgorithm");
646
647     else if (handoverAlg == "a3")

```

```

648     {
649         lteHelper->SetHandoverAlgorithmType("ns3::
           A3RsrpHandoverAlgorithm");
650         lteHelper->SetHandoverAlgorithmAttribute("Hysteresis
           ", DoubleValue(3.0));
651         lteHelper->SetHandoverAlgorithmAttribute("
           TimeToTrigger",
652           TimeValue(Milliseconds(256)));
653     }
654
655     else if (handoverAlg == "a2a4")
656     {
657         lteHelper->SetHandoverAlgorithmType("ns3::
           A2A4RsrqHandoverAlgorithm");
658         lteHelper->SetHandoverAlgorithmAttribute("
           ServingCellThreshold",
659           UIntegerValue(30));
660         lteHelper->SetHandoverAlgorithmAttribute("
           NeighbourCellOffset",
661           UIntegerValue(2));
662     }
663
664     else
665     {
666         NS_LOG_ERROR("Invalid handover algorithm option.");
667     }
668
669     //-----Cria o do RemoteHost
670     // Cria um simples RemoteHost
671     NodeContainer remoteHostContainer;
672     remoteHostContainer.Create(node_remote);
673     Ptr<Node> remoteHost = remoteHostContainer.Get(0);
674
675     // Pilha de Internet
676     InternetStackHelper internet;
677     internet.Install(remoteHost);
678
679     // Cria link Internet
680     PointToPointHelper p2ph;
681     p2ph.SetDeviceAttribute("DataRate", DataRateValue(
           DataRate("100Gb/s")));
682     p2ph.SetDeviceAttribute("Mtu", UIntegerValue(1400));
683     p2ph.SetChannelAttribute("Delay", TimeValue(Seconds
           (0.010)));
684     NetDeviceContainer internetDevices = p2ph.Install(pgw,
           remoteHost);
685
686     // Determina endere o ip para o Link
687     Ipv4AddressHelper ipv4h;
688     ipv4h.SetBase("10.1.0.0", "255.255.0.0");
689     Ipv4InterfaceContainer internetIpIfaces;
690     internetIpIfaces = ipv4h.Assign(internetDevices);
691
692     // interface 0 localhost e interface 1 dispositivo

```

```

p2p
693 Ipv4Address remoteHostAddr = internetIpIfaces.GetAddress
      (1);
694 Ipv4StaticRoutingHelper ipv4RoutingHelper;
695 Ptr<Ipv4StaticRouting> remoteHostStaticRouting =
      ipv4RoutingHelper.GetStaticRouting(remoteHost->
      GetObject<Ipv4>());
696 remoteHostStaticRouting->AddNetworkRouteTo(Ipv4Address("
      7.0.0.0"),
697 Ipv4Mask("255.0.0.0"), 1);
698
699 NodeContainer nodes_ue_nc;
700 nodes_ue_nc.Create(node_ue);
701
702 NodeContainer cbr_nodes;
703 cbr_nodes.Create(n_cbr);
704
705 NodeContainer enbNodes;
706 enbNodes.Create(enb_HP_N + low_power + hot_spot);
707
708 internet.Install(nodes_ue_nc);
709 internet.Install(cbr_nodes);
710
711 /*----- POSI  O DAS TORRES -----*/
712 Ptr<ListPositionAllocator> HpnPosition = CreateObject<
      ListPositionAllocator>();
713 ArrayPositionAllocator(HpnPosition);
714
715 MobilityHelper remoteHostMobility;
716 remoteHostMobility.SetMobilityModel("ns3::
      ConstantPositionMobilityModel");
717 remoteHostMobility.Install(remoteHost);
718 remoteHostMobility.Install(pgw);
719
720 /*----- MONILIDADE DAS TORRES -----*/
721
722 MobilityHelper mobilityEnb;
723 mobilityEnb.SetMobilityModel("ns3::
      ConstantPositionMobilityModel");
724 mobilityEnb.SetPositionAllocator(HpnPosition);
725 mobilityEnb.Install(enbNodes);
726
727 MobilityHelper mobilityCbr;
728 mobilityEnb.SetMobilityModel("ns3::
      ConstantPositionMobilityModel");
729 mobilityEnb.SetPositionAllocator(HpnPosition);
730 mobilityEnb.Install(cbr_nodes);
731
732 Ns2MobilityHelper ue_mobil = Ns2MobilityHelper("mobil/
      SanFrancisco.tcl");
733 MobilityHelper ueMobility;
734 MobilityHelper enbMobility;
735
736 ue_mobil.Install(nodes_ue_nc.Begin(), nodes_ue_nc.End())

```

```

737     ;
738     //-----Instala LTE Devices para cada grupo de
739     n s
740     NetDeviceContainer enbLteDevs;
741     enbLteDevs = lteHelper->InstallEnbDevice(enbNodes);
742     NetDeviceContainer pedLteDevs;
743     NetDeviceContainer carLteDevs;
744     carLteDevs = lteHelper->InstallUeDevice(nodes_ue_nc);
745     NetDeviceContainer cbrLteDevs;
746     cbrLteDevs = lteHelper->InstallUeDevice(cbr_nodes);
747
748     /*-----*/
749     Ipv4InterfaceContainer pedIpIface;
750     pedIpIface = epcHelper->AssignUeIpv4Address(
751         NetDeviceContainer(pedLteDevs));
752     Ipv4InterfaceContainer carIpIface;
753     carIpIface = epcHelper->AssignUeIpv4Address(
754         NetDeviceContainer(carLteDevs));
755     Ipv4InterfaceContainer cbrIpIface;
756     cbrIpIface = epcHelper->AssignUeIpv4Address(
757         NetDeviceContainer(cbrLteDevs));
758
759     for (uint32_t u = 0; u < nodes_ue_nc.GetN(); ++u)
760     {
761         Ptr<Node> ueNode = nodes_ue_nc.Get(u);
762         Ptr<Ipv4StaticRouting> ueStaticRouting =
763             ipv4RoutingHelper.GetStaticRouting(ueNode->
764                 GetObject<Ipv4>());
765         ueStaticRouting->SetDefaultRoute(epcHelper->
766             GetUeDefaultGatewayAddress(),
767             1);
768         // Simulator::Schedule(Seconds(0.0), &showPosition,
769             ueNode, 1.0);
770     }
771
772     /*-----CONFIGURA O DE CBR-----*/
773     uint16_t cbrPort = 3000;
774     ApplicationContainer clientApps;
775     ApplicationContainer serverApps;
776
777     //-----DEFINIR GATEWAY-----
778     for (uint32_t u = 0; u < cbr_nodes.GetN(); ++u)
779     {
780         Ptr<Node> ueNode = cbr_nodes.Get(u);
781         Ptr<Ipv4StaticRouting> ueStaticRouting =
782             ipv4RoutingHelper.GetStaticRouting(ueNode->
783                 GetObject<Ipv4>());
784         ueStaticRouting->SetDefaultRoute(epcHelper->
785             GetUeDefaultGatewayAddress(),
786             1);
787         Ptr<Ipv4> ipv4 = cbr_nodes.Get(u)->GetObject<Ipv4>()

```

```

779     ;
780     Ipv4InterfaceAddress iaddr = ipv4->GetAddress(1, 0);
781     Ipv4Address addri = iaddr.GetLocal();
782     PacketSinkHelper packetSinkHelper(
783         "ns3::UdpSocketFactory",
784         InetSocketAddress(Ipv4Address::GetAny(), cbrPort
785             ));
786     serverApps.Add(packetSinkHelper.Install(cbr_nodes.
787         Get(u)));
788     serverApps.Start(Seconds(20));
789
790     int load = 1024;
791     UdpClientHelper client(addri, cbrPort);
792     client.SetAttribute("Interval",
793         TimeValue(MilliSeconds(
794             interPacketInterval)));
795     client.SetAttribute("MaxPackets", UintegerValue
796         (100000000));
797     client.SetAttribute("PacketSize", UintegerValue(load
798         ));
799     NS_LOG_INFO("C lula " << u + 1 << " com cbr de " <<
800         load << " bytes" << std::endl);
801
802     clientApps.Add(client.Install(remoteHost));
803
804     clientApps.Start(Seconds(20));
805 }
806
807 /*----POTENCIA DE TRSMISSAO-----*/
808 Ptr<LteEnbPhy> enb0Phy;
809
810 // todo: set different bandwidth for individual cells
811 for (int i = 0; (unsigned)i < enbLteDevs.GetN(); i++)
812 {
813     enb0Phy = enbLteDevs.Get(i)->GetObject<
814         LteEnbNetDevice>()->GetPhy();
815     if (i < enb_HPN)
816     {
817         enb0Phy->SetTxPower(hpnTxPower);
818     }
819     else if (i < enb_HPN + low_power)
820     {
821         enb0Phy->SetTxPower(lpnTxPower);
822     }
823     else if (i < enb_HPN + low_power + hot_spot)
824     {
825         enb0Phy->SetTxPower(lpnTxPower);
826     }
827 }
828
829 //-----Anexa as UEs na eNodeB
830
831 lteHelper->Attach(carLteDevs);
832 lteHelper->AttachToClosestEnb(cbrLteDevs, enbLteDevs);

```

```

825     lteHelper->AddX2Interface(enbNodes);
826
827     NS_LOG_INFO("Create Applications.");
828
829     // Início Transmissão de Vídeo
830     //Rodar aplicação EvalVid
831     requestStream(remoteHost, nodes_ue_nc, remoteHostAddr,
832                  simTime, transmissionStart);
833
834     /*-----NETANIM-----*/
835     AnimationInterface anim("v2x_temp/LTEnormal_v2x.xml");
836     // Cor e Descrição para eNb
837     for (uint32_t i = 0; i < enbNodes.GetN(); ++i)
838     {
839         anim.UpdateNodeDescription(enbNodes.Get(i), "eNb");
840         anim.UpdateNodeColor(enbNodes.Get(i), 0, 255, 0);
841     }
842     for (uint32_t i = 0; i < nodes_ue_nc.GetN(); ++i)
843     {
844         anim.UpdateNodeDescription(nodes_ue_nc.Get(i), "UE
845         Carro");
846         anim.UpdateNodeColor(nodes_ue_nc.Get(i), 255, 0, 0);
847     }
848     for (uint32_t i = 0; i < cbr_nodes.GetN(); ++i)
849     {
850         anim.UpdateNodeDescription(cbr_nodes.Get(i), "CBR");
851         anim.UpdateNodeColor(cbr_nodes.Get(i), 0, 255, 0);
852     }
853     anim.UpdateNodeDescription(remoteHost, "RH");
854     anim.UpdateNodeColor(remoteHost, 0, 255, 255);
855
856     Simulator::Stop(SIMULATION_TIME_FORMAT(simTime));
857
858     /*-----NOTIFICAÇÕES DE HANDOVER E SINAL-----*/
859     Config::Connect("/NodeList/*/DeviceList/*/LteUeRrc/
860     ConnectionEstablished",
861                    MakeCallback(&
862                                NotifyConnectionEstablishedUe));
863     Config::Connect("/NodeList/*/DeviceList/*/LteUeRrc/
864     HandoverStart",
865                    MakeCallback(&NotifyHandoverStartUe));
866     Config::Connect("/NodeList/*/DeviceList/*/LteUeRrc/
867     HandoverEndOk",
868                    MakeCallback(&NotifyHandoverEndOkUe));
869
870     //lteHelper->EnableTraces();
871
872     Simulator::Run();
873     Simulator::Destroy();
874
875     return EXIT_SUCCESS;
876 }

```

Listing 1: Código Fonte do Algoritmo Skip-Hove escrito na linguagem C