

PE-MHMR: Previsão de Rotas de Múltiplos Saltos e Energeticamente Eficientes para Descarregamento de Dados em Redes Veiculares

Efrem Eladie de Oliveira Lousada¹, Fátima Duarte-Figueiredo²

¹Instituto Federal de Minas Gerais – Ibirité, MG – Brazil

²Programa de Pós-Graduação em Informática – PUC Minas – Brazil

efrem.lousada@ifmg.edu.br, fatima.duarterfigueiredo@gmail.com

Abstract. *Vehicular Ad-Hoc Networks (VANETs) and Intelligent Transportation Systems (ITS) have experienced remarkable developments in the last years. Finding the optimal route for data offloading from vehicles can ensure the effectiveness of ITSs and the network. This work presents the Predicted Energetic Multi-hop Multi-RSU (PE-MHMR) for data offloading in VANETS, considering multiple hops in RSU (Radio Side Units) overlapping scenarios. It is based on an efficient transfer from the source node to the destination node, by finding the shortest and most energy-efficient path. Complex networks were used to calculate the links probability success. Shorter, more energy-efficient and more successful routes are sought by the PE-MHMR algorithm. The metrics used to evaluate the algorithm were data loss, offloading fraction and successful offloading fraction. The simulation results demonstrate that more efficient routes can be calculated using PE-MHMR.*

Resumo. *As Redes Veiculares Ad-Hoc (Vehicular Ad-Hoc Networks - VANETs) e os Sistemas de Transporte Inteligente (Intelligent Transportation Systems - ITSs) tiveram um desenvolvimento surpreendente, nos últimos anos. Encontrar a rota ideal para descarregamento de dados (offloading) de veículos pode garantir a eficácia dos ITSs e da rede. Este trabalho apresenta o Predicted Energetic Multi-hop Multi-RSU (PE-MHMR) para descarregamento de dados em VANETS, considerando múltiplos saltos em cenários de sobreposição de RSUs (Radio Side Units). Ele se baseia em uma transferência eficiente do nó de origem para o nó de destino, através da descoberta do caminho mais curto e energeticamente mais eficiente. Redes complexas foram utilizadas para o cálculo de probabilidade de sucesso dos enlaces. Rotas mais curtas, mais energeticamente eficientes e com maior probabilidade de sucesso são buscadas pelo algoritmo PE-MHMR. As métricas utilizadas para avaliar o algoritmo foram perda de dados, fração de descarregamento e fração de descarregamento bem-sucedido. Os resultados das simulações demonstram que rotas mais eficientes podem ser calculadas utilizando o PE-MHMR.*

1. Introdução

Com a crescente demanda por serviços que utilizam redes veiculares e a grande quantidade de dados gerados pelos veículos, é necessário desenvolver estratégias para a

eficácia dessas redes. Durante o processo de descarga de dados (*offloading*), os dados de um veículo são transmitidos para uma RSU (*Road Side Unit*) através de comunicações Veículo-a-Veículo (V2V) e Veículo-a-Infraestrutura (V2I) [Hussein and Mahmood 2023]. A descarga de dados V2V e V2I é um problema importante a ser resolvido nas VANETs. O uso de servidores MEC (*Mobile Edge Computing*) pode reduzir a carga da rede e melhorar o desempenho da descarga de dados [Islam et al. 2023]. No entanto, vários desafios envolvem essa tecnologia [Hassan et al. 2021]. Durante a descarga da rede veicular para a infraestrutura, pode surgir a necessidade de roteamento *multi-hop* para garantir que a informação chegue à RSU [Bello Tambawal et al. 2019]. A solução deve escolher o melhor caminho entre o veículo que precisa descarregar dados e a RSU. Durante a seleção do caminho de transmissão ideal, fatores como consumo de energia, probabilidade de sucesso na entrega e tempo de entrega devem ser considerados [Zhu et al. 2023].

Este trabalho é derivado de uma tese de doutorado [Lousada and Duarte-Figueiredo 2024], [Lousada and Figueiredo 2024a] e [Lousada and Figueiredo 2024b] e apresenta uma solução para descarga de dados considerando múltiplos saltos e sobreposição de RSUs. O algoritmo PE-MHMR (*Predicted Energetic Multi-hop Multi-RSU*) encontra o servidor MEC mais eficiente para a descarga veicular. Isso é feito avaliando a aptidão das RSUs intermediárias para fornecer uma comunicação veicular ininterrupta. Essas avaliações são feitas de forma contínua, até que uma rota de alta probabilidade de sucesso e energeticamente eficiente seja encontrada para a transferência eficiente entre os nós da rede. O principal objetivo do algoritmo é encontrar, para o descarregamento, a melhor rota envolvendo múltiplos saltos e múltiplas RSUs. Perda de dados, fração de descarregamento e fração de descarregamento bem-sucedido são resultados de simulação que apontam que o PE-MHMR é eficaz para o descarregamento em redes veiculares com múltiplas RSUs e múltiplos saltos.

O artigo está organizado da seguinte maneira: Na seção 2, são apresentados os trabalhos relacionados. Na seção 3, é mostrada a nossa proposta, o PE-MHMR. Na seção 4, as simulações e os resultados são apresentados. Por fim, a seção 5 conclui e propõe trabalhos futuros.

2. Trabalhos Relacionados

Em cenários onde a latência é um fator crítico — como em jogos online, realidade aumentada e redes veiculares — o descarregamento de dados, ou *offloading*, é uma solução eficiente para atender aos requisitos de qualidade de serviço (QoS - *Quality of Service*) [Duarte-Figueiredo and Loureiro 2004] das aplicações [Feng et al. 2022].

Em [Huang et al. 2020], foi proposto um método para descarregamento de dados veiculares, chamado de seleção de caminho de descarregamento VVR, baseado no limite de tempo de retransmissão do destino, que utiliza uma abordagem de Computação de Borda Móvel (MEC) em vez do método tradicional de descarregamento V2I. O método proposto visa realizar o descarregamento de dados usando a RSU X, n saltos de distância, selecionando o caminho de descarregamento de dados VVR mais adequado. O servidor MEC recebe relatórios de contexto periódicos dos veículos e utiliza mecanismos de detecção e redução (DAS), detecção e extensão (DAE) e recuperação de caminho para construir e manter o caminho de descarregamento de dados VVR.

Os autores de [Huang and Lai 2020] propuseram um Método de Construção de Caminho de Descarregamento de Dados VVR baseado em Utilidade Limitada e de k -saltos com Restrição de Atraso (DC-KUPC).

O artigo [Song et al. 2023b] propõe o método de "*Predicted Utility-centric Path Switching*" (PUPS) para descarregamento de dados em múltiplos saltos (k -hop) no contexto de comunicações V2V, utilizando um controlador de redes definidas por software (SDN) dentro da arquitetura de computação de borda de acesso múltiplo (MEC). O objetivo é melhorar a eficiência do descarregamento de dados entre veículos que geralmente se comunicam através de estações rádio-base (BSs) de redes 4G/5G.

O artigo [Choi and Lim 2023] propõe um método de descarregamento de tarefas e agendamento de trabalhos baseado em computação em nuvem e borda para sistemas IoV (Internet of Vehicles). Ele visa melhorar a qualidade do serviço ao reduzir a latência e o congestionamento. Utilizando informações de localização e comunicações V2V (veículo a veículo), o método otimiza a posição de descarregamento e usa a otimização por colônia de formigas (ACO) para encontrar a melhor estratégia. O método não considera múltiplas RSUs.

A otimização conjunta de descarregamento de tarefas, alocação de recursos e *caching* de dados em uma rede veicular assistida por MEC (*Mobile Edge Computing*) é apresentada em [Dai 2023]. O problema é formulado como um *Mixed-Integer Non-Linear Programming* (MINLP) e é resolvido utilizando um algoritmo de otimização por enxame de partículas binárias (BPSO).

Uma estratégia centralizada para o posicionamento de RSUs em redes veiculares é apresentada em [Rashid et al. 2023]. O foco é monitorar a densidade do tráfego de veículos usando parâmetros a taxa de transmissão e localização. A otimização do descarregamento de tarefas e *cache* de conteúdo no contexto da Internet dos Veículos (IoV) é mostrada em [Wang et al. 2023].

Em [Song et al. 2023a], os autores propõem um algoritmo de descarregamento de tarefas baseado na otimização dingo (DDOA) em redes multihop habilitadas para comunicação V2V e V2I. Os autores constroem uma arquitetura de computação de borda com comunicações multihop V2V/V2I, onde veículos e RSUs colaboram para ampliar os recursos de comunicação.

O método chamado Predicted K-hop-limited Multi-RSU-considered (PKMR) para descarregamento de dados de unidade veicular-para-veicular-para-RSU (VVR) em um ambiente de computação de borda multiacesso (MEC) utiliza um controlador de Rede Definida por Software (SDN) [Da-Silva et al. 2019] dentro do servidor MEC para gerenciar o processo de descarregamento. Considerando os caminhos previstos e as condições de rede dos veículos e das unidades de borda (RSUs), o método PKMR seleciona o caminho de descarregamento de dados VVR mais adequado. A avaliação de desempenho mostra que o PKMR supera os métodos tradicionais de auto-descarregamento. O método proposto aborda desafios relacionados à implantação de múltiplas RSUs, sobreposição de sinais de RSU e descarregamento de dados [Lin et al. 2022]. No entanto, não considera o consumo de energia para calcular a melhor rota, nem o tempo necessário para a transmissão dos dados, tornando-o inadequado para soluções em tempo real.

O trabalho apresentado em [Zeng et al. 2024] aborda a complexidade do descar-

regamento de tarefas em um cenário de veículos conectados, onde a movimentação dos veículos resulta na troca de RSUs e servidores. Essa dinâmica pode gerar falhas no descarregamento de tarefas ou aumentar os atrasos nos serviços oferecidos. Para enfrentar esses desafios, os autores propõem um modelo preditivo baseado em uma rede neural Bi-LSTM para antecipar as trajetórias dos veículos. Essa abordagem se traduz na divisão da área de serviço em grades de tamanhos iguais, permitindo que a precisão das previsões seja avaliada de forma mais eficaz.

Alguns dos trabalhos relacionados utilizam o cenário tradicional V2I, o que significa descarregamento considerando apenas um salto. Alguns dos trabalhos focaram apenas no descarregamento através de servidores MEC. Abordagens alternativas na exploração e definição das áreas críticas são reduzidas a regiões retangulares, o que não é aplicável em alguns cenários. Além disso, uma abordagem estendida pode ser feita para encontrar outras aplicações para melhorar a viabilidade do conteúdo e reduzir o tráfego de retorno. Os trabalhos apresentados para múltiplas RSUs não consideram o consumo de energia nem o tempo necessário para a transmissão dos dados, tornando-os impraticáveis para soluções em tempo real.

3. PE-MHMR

O PE-MHMR (*Predicted Energetic Multi-hop Multi-RSU*) é uma solução algorítmica para descarga de dados, em redes veiculares, considerando múltiplos saltos e múltiplas RSUs. O PE-MHMR traça rotas entre um veículo que vai descarregar dados e uma RSU. A ação de encaminhamento é denominada VVR (Veículo para Veículo para RSU) [Lin et al. 2022]. Nela, os dados são transferidos de nós veículos para outros nós veículos, até chegarem em uma RSU. Da RSU, os dados são encaminhados ao veículo destino, onde acontece o descarregamento de dados (*offloading*). A melhor rota é calculada com base na avaliação externa da qualidade do sinal da RSU, na estimativa de energia e na qualidade de enlace entre veículos, calculada usando métricas de rede complexas. A probabilidade de retransmissão com sucesso é calculada para cada nó candidato a fazer parte do caminho VVR. Escolhida a melhor rota, o descarregamento é realizado: um nó transmite para o próximo nó da rota definida. Caso ocorra algum problema durante a retransmissão, recalcula-se uma nova rota. Tudo isso acontece, no PE-MHMR, através da invocação de algoritmos que promovem cálculos e seleções que levam à escolha da melhor rota. Os algoritmos e as equações dos cálculos envolvidos serão detalhados adiante.

O PE-MHMR comprehende quatro estados, descritos a seguir. O único estado obrigatório é o de inicialização. Os demais ocorrem em situações específicas, devido à constante movimentação de veículos, implicando em mudanças de tráfego e rota. Para a execução de cada estado, no PE-MHMR, fórmulas e algoritmos de funções parciais foram propostos. A meta deles é a escolha dos melhores enlaces para a transmissão de dados entre o Vs (*vehicle source*) e a RSU. Os melhores enlaces são os que apresentam maior disponibilidade energética, menor atraso médio, maior taxa de entrega e de precisão, além da menor probabilidade de recálculo de rota.

No estado de inicialização, um veículo agente (V_s *Vehicular source*) pode se comunicar com um veículo par (V_p - *Vehicular pair*), tendo como objetivo descarregar dados em uma RSU. Periodicamente, os veículos enviam uma mensagem de contexto ao servidor MEC. Quando o servidor MEC recebe o contexto relatado pelo veículo V_s , no

tempo T_c , ele é acionado para determinar se existem caminhos de descarregamento V2V2I (*Vehicle-to-Vehicle-to-Infrastructure*) para o veículo V_s . Se o servidor MEC encontrar um caminho de descarregamento V2V2I para o V_s , ele calcula todos os possíveis caminhos V2V2I que possam existir durante o intervalo $[T_c, T_c + t]$ e seleciona o melhor caminho de descarregamento V2V2I, gerado no tempo T_0 . O instante T_0 está dentro do intervalo $[T_c, T_c + t]$. Subsequentemente, o servidor MEC envia mensagens aos veículos constituintes do caminho de descarregamento V2V2I, para ativar a sessão de descarregamento V2V2I no tempo T_0 . Durante a sessão de descarregamento, todos os veículos constituintes do caminho de descarregamento V2V2I relatam seus contextos ao MEC, alinhando o tempo local e o GPS (*Global Positioning System*). Após receber os contextos, o MEC calcula o tempo de cada enlace V2V e o enlace V2I, representando o tempo que o veículo de descarregamento fica dentro da cobertura do sinal da RSU.

O Algoritmo 1 calcula o tempo de início e de término para todos os enlaces. Esse cálculo é baseado na velocidade de localização de cada par de nós (p, q) no ambiente (*NodeSet*), na geo-distância, na faixa de sinal da unidade de bordo e na função de adequação para energia, conforme proposto em [Lin et al. 2022]. São consideradas a posição e a velocidade de cada par de veículos p e q , onde tanto p quanto q fazem parte do *NodeSet* no tempo t_0 . As posições podem ser representadas pelas coordenadas $Pos_x(p)$, $Pos_y(p)$ e $Pos_x(q)$, $Pos_y(q)$, e as velocidades por $Vel_x(p)$, $Vel_y(p)$ e $Vel_x(q)$, $Vel_y(q)$. Os veículos podem se comunicar em um raio R de 300m, assim como no IEEE 802.11p OBU.

Algoritmo 1: Encontre Todos os Intervalos de Tempo de Conexão dos enlaces

```

1 for each enlace  $(p, q)$  do
2   
$$(Px(p) - Px(q) + t(Vx(p) - Vx(q)))^2 + (Py(p) - Py(q) + t(Vy(p) - Vy(q)))^2 - R^2 = 0$$


```

Uma vez que os pontos de início e fim de cada enlace são identificados, é calculado o caminho adequado para o *offloading*. Para cada enlace, em cada caminho, é calculada uma função de probabilidade de sucesso (SP), representada pela equação 1.

$$SP = Ql \times Ff \times Cp \quad (1)$$

Na equação 1, SP significa sucesso na transmissão, Ql representa qualidade do link, Ff função de aptidão e Cp probabilidade de sucesso na transmissão. A qualidade leva em consideração: (i) tempo de vida mais longo do caminho V2V2I; (ii) se houver mais veículos dentro da cobertura de sinal de um veículo constituinte, pode haver potenciais colisões de transmissão, levando a um valor de utilidade mais baixo devido aos valores de espera maiores necessários para o acesso ao canal; (iii) o número total de veículos dentro da cobertura de sinal de cada veículo constituinte influencia no valor de utilidade do caminho, indicando que um caminho mais congestionado resulta em um valor de utilidade mais baixo. Uma função de qualidade que pode derivar a qualidade do caminho de descarregamento de dados VVR é definida da seguinte forma, cujas variáveis utilizadas são definidas e explicadas na Tabela 1. Ql é baseado em [Huang and Lai 2020].

$$Q_l = \frac{T^{\text{offloading}}}{\alpha(2^{\lfloor \log_2 N^{\max} \rfloor} + N^{\max}) + (1 - \alpha) N^{\text{total}}} \cdot \frac{1}{\frac{\sqrt{\sum_{i=1}^n i^2}}{n}} \quad (2)$$

Table 1. Variáveis da Função de Qualidade

Parameter	Description
$T_{\text{offloading}}$	O tempo de vida do caminho de descarregamento V2V2I, que é o tempo mínimo de conexão dos enlaces constituintes no caminho de descarregamento V2V2I.
N_i	O número de veículos que estão na cobertura de sinal do veículo constituinte do caminho de descarregamento V.
N_{\max}	Max (N1, N2, ..., Nn).
N_{total}	Soma de (N1, N2, ..., Nn).
n	A contagem de saltos no caminho de descarregamento V2V2I, incluindo o enlace V2I do agente de descarregamento para RSU.

Fonte: [Huang and Lai 2020]

A função de aptidão (Ff), definida pela equação 3, avalia o valor de aptidão da posição de um veículo em um espaço 2D em relação ao seu par. Ela indica a qualidade de um nó na rede, combinando métricas de eficiência de comunicação e penalidades energéticas. O objetivo é encontrar um equilíbrio entre conectividade e eficiência energética, com impacto direto na manutenção e na durabilidade da rede. A fórmula considera tanto fatores espaciais quanto conectividade e energia. Os pesos α_1 à α_3 são ajustáveis, dependendo da aplicação.

$$\text{fitness_val}_i \leftarrow \alpha_1 \cdot x1_i + \alpha_2 \cdot x2_i + \alpha_3 \cdot x3_i - \text{penalties}_i \quad (3)$$

onde α_1 , α_2 e α_3 são pesos atribuídos a cada métrica para refletir a importância relativa de cada uma, permitindo ajuste fino de acordo com os objetivos da rede (priorizar alcance, estabilidade, etc.) e penalties_i é uma penalidade associada ao nó n_i , aplicada em função do nível de energia remanescente. As variáveis $x1_i$, $x2_i$ e $x3_i$ são definidas a seguir.

A variável $x1_i$, definida na equação 4, calcula a distância média de n_i para seus vizinhos diretos Vn_i . d_{ij} representa a distância entre n_i e cada nó vizinho j .

$$x1_i \leftarrow \frac{\sum_{j \in \{Vn_i\}} d_{ij}}{Cn_i} \quad (4)$$

A variável $x2_i$, representada pela equação 5, calcula a energia média dos nós vizinhos de n_i , sendo e_j a energia de cada vizinho j . Esse cálculo mede a “saúde” energética da vizinhança.

$$x2_i \leftarrow \frac{\sum_{j \in \{Vn_i\}} e_j}{Cn_i} \quad (5)$$

A variável $x3_i$ representada pela equação 6, é a proporção de nós que n_i pode alcançar em sua vizinhança Vn_i através de rotas de múltiplos saltos.

$$x3_i \leftarrow \frac{\text{número de nós acessíveis via rotas de múltiplos saltos a partir de } n_i}{Cn_i} \quad (6)$$

A penalidade é adicionada para nós com energia crítica. As regras de penalização para nós com menos energia são: penalidade de 80% para energia abaixo de 50%, de 85% para energia abaixo de 40%, de 90% para energia abaixo de 30%, de 95% para energia abaixo de 20%, de 97,5% para energia abaixo de 10%, de 99% para energia abaixo de 5%.

$\text{penalty}_i \leftarrow \text{penalidade se a energia de } n_i \text{ cair abaixo do limite mínimo de 50\%}.$

A eficiência de um enlace é determinada por uma função de probabilidade detalhadamente descrita em artigo prévio dos autores deste trabalho [Lousada et al. 2019]. Essa função calcula a probabilidade de sucesso da transmissão (C_p), formulada pela Equação (7), que é comparada a um valor de limiar derivado de uma média móvel exponencialmente ponderada. Se C_p exceder o limiar, o enlace é penalizado e pode ser classificado mais baixo. O valor de C_p é calculado pelo produto da probabilidade associada ao Grau do Vértice (pGV), da probabilidade associada à Sobreposição Topológica (pST) e da probabilidade associada à Persistência da Aresta (pPA), formulado pela Equação (7).

$$C_p = pGV \times pST \times pPA \quad (7)$$

Com a função de probabilidade de sucesso definida, os caminhos candidatos para descarregamento podem ser determinados. O algoritmo 2 (`InitContenderSet`) tem como objetivo inicializar o conjunto de caminhos candidatos para encontrar o melhor caminho de V_{root} para outros nós dentro de um conjunto de nós (`NodeSet`) e um conjunto de pontos no tempo (`TimeSet`). Primeiramente, o conjunto de caminhos candidatos (`ContenderSet`) é inicializado como vazio. Em seguida, há uma iteração, no algoritmo, entre cada nó v e o `NodeSet`, no tempo t , no `TimeSet`. Durante a iteração, o algoritmo verifica se existe um enlace ativo entre V_{root} e o nó v no tempo t . Se tal enlace existir, um novo caminho candidato é criado, começando em V_{root} e terminando em v , no tempo t . O novo caminho candidato é, então, atribuído. As suas informações relevantes incluem o nó de início, o nó final, o tempo adiado, o tempo de vida, a qualidade do caminho calculada usando uma função específica e a contagem de saltos, que é definida como 1, pois é um caminho de um salto de V_{root} para v . Este caminho candidato é adicionado ao conjunto de caminhos candidatos (`ContenderSet`), para consideração posterior. Após completar as iterações entre nós e pontos, no tempo, o algoritmo retorna o `ContenderSet` contendo todos os possíveis caminhos de V_{root} para os nós no `NodeSet` que estão ativos em seus respectivos pontos de tempo. Esse conjunto será utilizado nos processos subsequentes para determinar o melhor caminho de V_{root} para o nó final (RSU).

Algoritmo 2: InitContenderSet

```
1 ContenderSet  $\leftarrow \emptyset$ 
2 linksToVroot  $\leftarrow$  Links from Vroot to NodeSet
3 for each  $v$  in NodeSet do
4   for each  $t$  in TimeSet do
5     for each link  $(p, q)$  in linksToVroot do
6       if  $t_{start}(link) \leq t$  and  $t_{end}(link) > t$  then
7         NewContenderPath  $\leftarrow$  Create new path from Vroot to  $v$ 
8         starting at  $t$ 
9         End(NewContenderPath)  $\leftarrow v$ 
10        PostponedTime(NewContenderPath)  $\leftarrow t$ 
11        Lifetime(NewContenderPath)  $\leftarrow t_{end}(link) - t$ 
12        Quality(NewContenderPath)  $\leftarrow$  ComputeQuality
13        (NewContenderPath)
14        HopCount(NewContenderPath)  $\leftarrow 1$ 
15        NewContenderPath to ContenderSet
16
17 Return ContenderSet
```

O Algoritmo 3 (CBOP - *Constrained Best-Offloading Path*) foi projetado para encontrar o melhor caminho de descarregamento. Ele cria todos os possíveis caminhos de um salto a partir do veículo raiz Vroot. Em seguida, entra em um *loop* (linha 4) que calcula a qualidade de todos os caminhos e seleciona o que tem o valor de qualidade máximo como Pathmax. O algoritmo retorna o caminho Pathmax, considerado o melhor.

Algoritmo 3: CBOP (Constrained Best-Offloading Path)

```
1 ContenderSet  $\leftarrow$  InitContenderSet (Vroot, NodeSet,
2 TimeSet)
3 Pathmax  $\leftarrow$  null
4 maxQuality  $\leftarrow -\infty$ 
5 while ContenderSet is not empty do
6   for each Path in ContenderSet do
7     quality  $\leftarrow$  calculateQuality (Path)
8     if quality  $\geq$  maxQuality then
9       maxQuality  $\leftarrow$  quality
10      Pathmax  $\leftarrow$  Path
11
12 if End (Pathmax) = RSU then
13   break
14 ContenderSet  $\leftarrow$  RefreshContenderSet (Pathmax,
15 ContenderSet, k)
16 ContenderSet.remove (Pathmax)
17
18 Return Pathmax
```

No estado de redução, para garantir que o tempo com restrição de atraso t não exceda a vida útil do sub-caminho fora-RSU, a restrição de atraso é definida para ser menor ou igual à vida útil do sub-caminho fora-RSU. Se um novo agente de descar-

regamento e/ou um caminho atualizado de descarregamento V2V2I não puder ser encontrado, o servidor MEC informará ao veículo V_s para retornar à rede celular, e a sessão de descarregamento será encerrada. Se o veículo V_s permanecer dentro da cobertura de sinal do RSU, indicando que o caminho de descarregamento ainda está intacto, o estado de redução é concluído, e o veículo V_s pode continuar com o descarregamento por conta própria.

No estado de auto-descarregamento, quando o veículo V_s está dentro da cobertura de sinal do RSU, ele pode se comunicar diretamente com seu V_p através do RSU. Durante esse estado, o servidor MEC calcula continuamente a duração de tempo que o veículo V_s permanece dentro da cobertura de sinal do RSU, com base no contexto periodicamente relatado pelo V_s . O servidor MEC recebe o contexto do V_s em um determinado ponto de tempo T_c , e sabe que o veículo V_s sairá da cobertura do RSU quando o próximo contexto for recebido, indicando que o veículo V_s está se afastando da cobertura de sinal do RSU. Essa transição desencadeia o início do estado de extensão.

No estado de extensão, quando o veículo fonte V_s , ou o agente de descarregamento do estado de extensão, sai da cobertura da RSU, o algoritmo de extensão é ativado. No tempo atual T_c , o servidor MEC recebe o contexto periodicamente relatado do agente de descarregamento e prevê que o veículo fonte ou o agente de descarregamento deixará a cobertura de sinal do RSU ao receber o próximo contexto. Esse evento desencadeia o algoritmo de extensão, que visa encontrar um novo agente de descarregamento e um caminho atualizado de descarregamento V2V2I. O servidor MEC avalia todos os candidatos potenciais que podem estar disponíveis durante o intervalo $[T_c, T_c + t]$. O caminho atual de descarregamento V2V2I do veículo V_s para o agente de descarregamento, que está dentro da cobertura de sinal do RSU, mas está saindo, é referido como sub-caminho c-offloading. Para garantir que o tempo com restrição de atraso t seja razoável e não exceda a vida útil do sub-caminho c-offloading, t é definido como menor ou igual à vida útil do sub-caminho c-offloading. Se o servidor MEC não conseguir encontrar um novo agente de descarregamento e/ou um caminho atualizado de descarregamento V2V2I, ele notificará o veículo V_s para retornar à rede celular e a sessão de descarregamento será encerrada.

Quando um caminho de descarregamento V2V2I sofre uma quebra inesperada, o algoritmo de recuperação de caminho é ativado para restaurar a conexão. Vamos considerar um sub-caminho $V_b - V_y - V_a$ dentro do caminho de descarregamento V2V2I para o veículo V_s . Se V_b não consegue transmitir dados para V_a , por exemplo, porque V_y sai da estrada, resultando na quebra do caminho de descarregamento V2V2I, denominamos este evento como o ponto de tempo quebrado T_c . O algoritmo de recuperação de caminho tenta encontrar um novo veículo V_x que possa reparar o caminho de descarregamento V2V2I, ou procura uma oportunidade para que V_b se reconecte com V_a dentro do intervalo $[T_c, T_c + t]$. Aqui, sub-caminho-a representa o sub-caminho de descarregamento atual de V_s para V_a , e sub-caminho-b-AD representa o sub-caminho de descarregamento atual de V_b para o agente de descarregamento. Para garantir que o comprimento de tempo com restrição de atraso t seja razoável e não exceda as vidas úteis de sub-caminho-a ou sub-caminho-b-AD, t é definido como menor ou igual à vida útil mínima entre sub-caminho-a e sub-caminho-b-AD. Se o algoritmo de recuperação de caminho falhar em reparar o caminho de descarregamento V2V2I, o servidor MEC informará

o veículo V_s para retornar à rede celular, e a sessão de descarregamento V2V2I será encerrada. O algoritmo utilizado para reconstrução, quando a RSU deve ser atualizada, é o mesmo proposto em [Lin et al. 2022].

4. Simulações e Resultados

A ferramenta SUMO (*Simulation of Urban Mobility*) foi utilizada para a simulação de mobilidade urbana. Dois simuladores foram utilizados junto ao SUMO, para simulação dos elementos e da comunicação entre eles: o NS-3 (*Network Simulator 3*) e o VANET Toolbox do MATLAB[Wang 2024].

Para o *PE-MHMR*, o VANET Toolbox do MATLAB foi utilizado. O modelo de tráfego adotado no trabalho foi criado através do SUMO. Foram simuladas três RSUs com sinais parcialmente sobrepostos e veículos se movimentando de forma aleatória, em diversas direções, em um plano, para simular saídas repentinas do alcance de RSUs e dos próprios veículos vizinhos, criando, assim, maior dificuldade na realização do *offloading*. O algoritmo Cp , que calcula a probabilidade de sucesso da transmissão, foi implementado através do NS-3, com o modelo de tráfego criado através do SUMO. O cenário simulado consiste em uma rodovia com 2 pistas e veículos se movendo na mesma direção, da esquerda para a direita. Para que veículos cruzem com outros veículos durante os seus trajetos, velocidades distintas foram adotadas para tornar possíveis as ultrapassagens. Detalhes de implementação e resultados comparativos do Cp com outros similares da literatura estão em nosso artigo [Lousada et al. 2019]. Neste artigo, o Cp foi invocado pelo PE-MHMR para o cálculo da probabilidade de sucesso da transmissão.

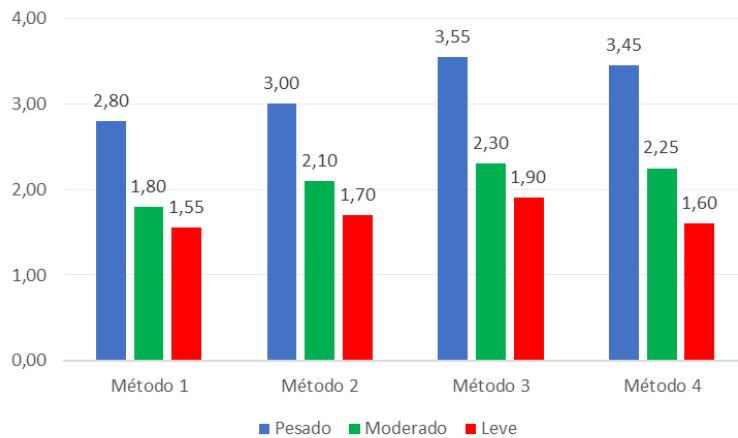
As densidades de veículos foram definidas e classificadas como trânsito leve (20 veículos/km), trânsito moderado (30 veículos/km), trânsito pesado (60 veículos/km) [Wisitpongphan et al. 2007]. Cada simulação foi executada com 300 segundos como tempo total. O intervalo de aquecimento da rede, conhecido como *warm-up*, é de 10 segundos.

Quatro métodos de *offloading* foram avaliados. O primeiro corresponde à nossa proposta completa, o PE-MHMR, oferecendo descarregamento de dados VVR multi-RSU com limitação de k saltos, considerando a qualidade do serviço, a energia e o mecanismo de previsão estendida no tempo. O segundo corresponde a um PE-MHMR incompleto, pois não considera a energia para o cálculo da rota e a probabilidade de entrega. O terceiro também corresponde a um PE-MHMR incompleto, pois não utiliza o mecanismo de previsão estendida no tempo. O quarto também corresponde a um PE-MHMR incompleto, pois não permite a troca de RSU se o link com a RSU for perdido.

Três métricas de desempenho são apresentadas: (I) perda de dados: proporção de dados perdidos no caminho de descarregamento de dados VVR, dividida pela quantidade total de dados transmitidos através das RSUs. (II) Fração de descarregamento de dados: proporção de dados descarregados que foram direcionados através das RSUs, dividida pela quantidade total de dados direcionados tanto pela BS quanto pelas RSUs. (III) Fração de descarregamento de dados bem-sucedida: proporção de dados que foram recebidos com sucesso pelo veículo fonte através do caminho de descarregamento de dados VVR, dividida pela quantidade total de dados transmitidos tanto pela BS quanto pelas RSUs.

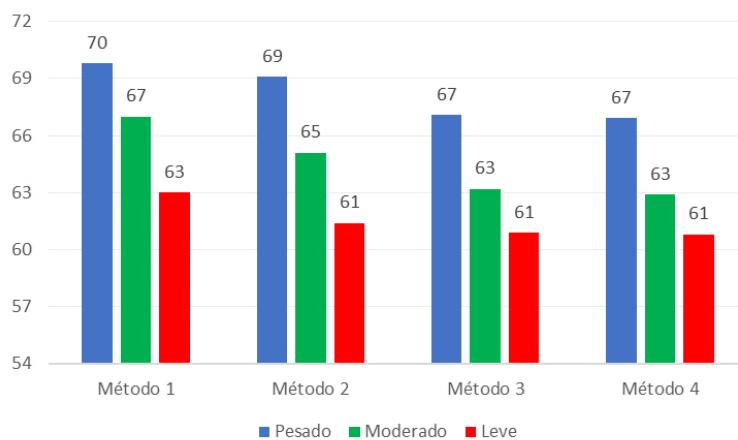
No gráfico da Figura 1, observa-se que a perda de dados no método 1 (PE-MHMR completo) é mais baixa que nos demais métodos. Os métodos que adotam o mecanismo de

Figure 1. Perda de Dados



previsão estendida no tempo em conjunto com a otimização de energia têm menos perdas. Isso ocorre porque o mecanismo de previsão estendida ajuda o servidor MEC a encontrar mais caminhos candidatos. Além disso, um caminho candidato que seja energeticamente melhor diminui os riscos de perda de comunicação devido a desconexões, aumentando a probabilidade de entrega bem-sucedida. Portanto, o caminho de descarregamento de dados VVR selecionado pode fornecer melhor qualidade de serviço (QoS) por ter menos perdas de dados. A perda de dados nos métodos 1 e 2 é ligeiramente menor do que nos métodos 3 e 4. Isso significa que métodos que permitem ao (Agente de Descarregamento) realizar a troca de RSU em regiões com sinais de RSU sobrepostos, ou seja, em ambientes com múltiplos RSUs, têm uma taxa de perda de dados mais baixa.

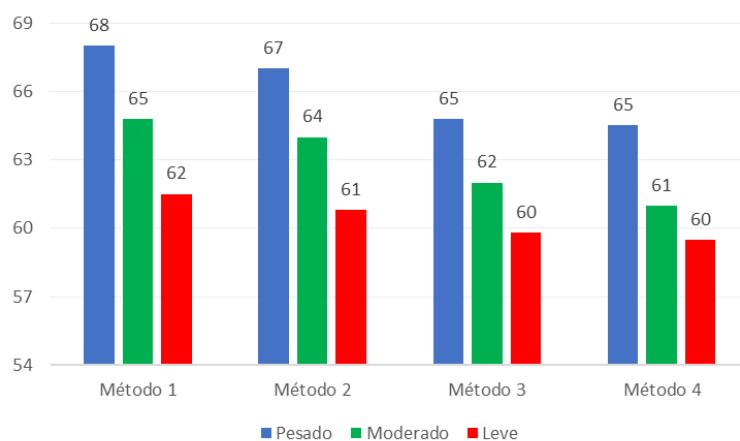
Figure 2. Fração de Descarregamento de Dados



O gráfico da Figura 2 ilustra a fração de descarregamento dados, ou *offloading*, que corresponde à quantidade de dados descarregados que foram direcionados através de RSUs dividida pela quantidade de dados que foram direcionados através de BSs e RSUs. A fração de descarregamento de dados em métodos que utilizam o mecanismo de previsão estendida no tempo associado a caminhos energeticamente eficientes é sempre maior. Essa relação se deve ao fato de o mecanismo de previsão estendida no tempo ofer-

ecer duas vantagens. A primeira vantagem é a utilização potencial de caminhos de descarregamento de dados VVR com vidas úteis mais curtas. Outra vantagem do mecanismo de previsão estendida no tempo é a capacidade de iniciar um caminho de descarregamento de dados mais cedo. Essa vantagem decorre da detecção de possíveis caminhos de descarregamento de dados VVR que podem não existir no momento em que o servidor MEC recebe o relatório de contexto periódico do veículo fonte, mas que se tornarão disponíveis no futuro, especificamente entre dois pontos de tempo consecutivos de relatórios de contexto. Nos métodos 1 e 2, tanto o AD (Agente de Descarregamento) quanto o veículo fonte podem utilizar a troca de RSU em regiões onde múltiplos RSUs têm sinais sobrepostos. Com isso, a fração de descarregamento de dados em um ambiente de múltiplos RSUs é maior em comparação a outros métodos nos quais o AD não realiza a troca de RSU. Essa relação ocorre porque, se o servidor MEC não consegue encontrar um caminho alternativo de descarregamento de dados VVR usando o esquema de re-seleção de AD, o método que incorpora a troca de RSU pode manter a sessão de descarregamento de dados VVR ao transitar para RSUs subsequentes ao longo do caminho original. Como resultado, permitir que tanto o AD quanto o veículo fonte realizem a troca de RSU estende a duração da sessão de descarregamento de dados VVR, resultando em uma maior fração de descarregamento de dados.

Figure 3. Fração de Offloading Bem-Sucedido



5. Conclusões e Trabalhos Futuros

Em um ambiente com múltiplas RSUs e veículos, a necessidade de soluções eficazes para reduzir a perda de pacotes e o tráfego de dados da rede é crucial. O PE-MHMR demonstrou ser uma abordagem inovadora e eficiente, mostrando que a consideração de parâmetros relacionados à qualidade do link entre veículos e ao fator de energia como entradas para a seleção do melhor caminho de tráfego de dados é essencial para o sucesso da comunicação. Até o momento, essas métricas haviam sido negligenciadas na maioria das abordagens existentes.

O PE-MHMR não apenas reduziu a taxa de perda de pacotes, mas também aumentou a taxa de sucesso no descarregamento de dados. Estes resultados destacam a eficiência do PE-MHMR em ambientes dinâmicos e complexos, onde as condições de tráfego e a qualidade dos links podem variar consideravelmente.

Para o futuro, há várias direções promissoras para expandir e aprimorar o PE-MHMR. A integração de novos modelos de mobilidade poderia fornecer uma compreensão mais profunda do comportamento dos veículos em diferentes cenários e melhorar a adaptação do método a novas condições de tráfego. Além disso, a adoção de tecnologias emergentes, como redes celulares 6G, poderia oferecer maior largura de banda e menores latências, beneficiando ainda mais o desempenho do PE-MHMR. Adicionalmente, o desenvolvimento de novos métodos para calcular a probabilidade de entrega bem-sucedida poderia refinar a função de ajuste do PE-MHMR, aprimorar a qualidade do link e aumentar a eficiência do link. Essas melhorias poderiam contribuir para a criação de redes veiculares mais robustas e confiáveis, capazes de suportar uma comunicação mais eficiente com qualidade de serviço.

References

- Bello Tambawal, A., Md Noor, R., Salleh, R., Chembe, C., and Oche, M. (2019). Enhanced weight-based clustering algorithm to provide reliable delivery for vanet safety applications. *PloS one*, 14(4):e0214664.
- Choi, Y. and Lim, Y. (2023). Deep reinforcement learning for edge caching with mobility prediction in vehicular networks. *Sensors*, 23(3):1732.
- Da-Silva, L. S., Storck, C. R., and de LP Duarte-Figueiredo, F. (2019). A dynamic load balancing algorithm for data plane traffic. In *IEEE Latin American Network Operations and Management Symposium, LANOMS*.
- Dai, W. (2023). Joint task offloading, resource allocation and data caching in mec-assisted vehicular network. In *2023 4th International Conference on Computer Engineering and Application (ICCEA)*, pages 70–76. IEEE.
- Duarte-Figueiredo, F. and Loureiro, A. A. F. (2004). *DiffMobil—Uma Arquitetura de Qualidade de Serviço Fima-Fim em Redes GPRS*. PhD thesis, UFMG.
- Feng, C., Han, P., Zhang, X., Yang, B., Liu, Y., and Guo, L. (2022). Computation offloading in mobile edge computing networks: A survey. *Journal of Network and Computer Applications*, 202:103366.
- Hassan, S.-U., Shabbir, M., Iqbal, S., Said, A., Kamiran, F., Nawaz, R., and Saif, U. (2021). Leveraging deep learning and sna approaches for smart city policing in the developing world. *International Journal of Information Management*, 56:102045.
- Huang, C.-M. and Lai, C.-F. (2020). The delay-constrained and network-situation-aware v2v2i vanet data offloading based on the multi-access edge computing (mec) architecture. *IEEE Open Journal of Vehicular Technology*, 1:331–347.
- Huang, C.-M., Lin, S.-Y., and Wu, Z.-Y. (2020). The k-hop-limited v2v2i vanet data offloading using the mobile edge computing (mec) mechanism. *Vehicular Communications*, 26:100268.
- Hussein, A. A. and Mahmood, D. A. (2023). Connectivity analysis in vehicular ad-hoc network based on vdtn. *Journal of Communications Software and Systems*, 19(2):147–157.
- Islam, M. M., Newaz, A. A. R., Song, L., Lartey, B., Lin, S.-C., Fan, W., Hajbabaie, A., Khan, M. A., Partovi, A., Phuapaiboon, T., et al. (2023). Connected autonomous vehi-

- cles: State of practice. *Applied Stochastic Models in Business and Industry*, 39(5):684–700.
- Lin, S.-Y., Huang, C.-M., and Wu, T.-Y. (2022). Multi-access edge computing-based vehicle-vehicle-rsu data offloading over the multi-rsu-overlapped environment. *IEEE Open Journal of Intelligent Transportation Systems*, 3:7–32.
- Lousada, E. E. and Duarte-Figueiredo, F. (2024). *UMA ABORDAGEM PARA OFFLOADING EM REDES VEICULARES CONSIDERANDO MÚLTIPLOS SALTOS EM UM CENÁRIO DE SOBREPOSIÇÃO DE RSU*. PhD thesis, PUC Minas.
- Lousada, E. E., Storck, C. R., Mini, R. A., and Duarte-Figueiredo, F. d. L. P. (2019). Using complex networks metrics to mitigate the broadcast storm problem. In *2019 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–6. IEEE.
- Lousada, E. E. d. O. and Figueiredo, F. d. L. P. D. (2024a). An approach for offloading with multi-hop considerations in an rsu signal overlay setting. *Revista de Gestão e Secretariado*, 15(4):e3739.
- Lousada, E. E. d. O. and Figueiredo, F. d. L. P. D. (2024b). Cn-ffvp: uma solução para mitigação de tempestade de broadcast baseada em métricas de redes complexas, distância e energia dos nos. *OBSERVATÓRIO DE LA ECONOMÍA LATINOAMERICANA*, 22(1):4494–4512.
- Rashid, S. A., Audah, L., Hamdi, M. M., Abdulsattar, N. F., Mutar, M. H., and Alkhafaji, M. A. (2023). Centralized rsu deployment strategy for effective communication in multi-hop vehicular adhoc networks (vanets). In *International Conference On Innovative Computing And Communication*, pages 53–65. Springer.
- Song, X., Wang, Y., Xie, Z., Zhang, R., and Xu, S. (2023a). Dingo-optimization-based task-offloading algorithm in multihop v2v/v2i-enabled networks. *Transactions on Emerging Telecommunications Technologies*, 34(9):e4823.
- Song, X., Zhang, R., Wang, Y., Yu, Y., and Xu, S. (2023b). Incentive mechanism design for two-layer mobile data offloading networks: a contract theory approach. *Ad Hoc Networks*, 144:103154.
- Wang, L. (2024). Vanet toolbox: A vehicular network simulator based on discrete event simulation. *MATLAB Central File Exchange*.
- Wang, P., Wang, Y., Qiao, J., and Hu, Z. (2023). Traffic-aware optimization of task offloading and content caching in the internet of vehicles. *Applied Sciences*, 13(24):13069.
- Wisitpongphan, N., Tonguz, O. K., Parikh, J. S., Mudalige, P., Bai, F., and Sadekar, V. (2007). Broadcast storm mitigation techniques in vehicular ad hoc networks. *IEEE Wireless Communications*, 14(6):84–94.
- Zeng, F., Zhang, Z., and Wu, J. (2024). Task offloading delay minimization in vehicular edge computing based on vehicle trajectory prediction. *Digital Communications and Networks*.
- Zhu, H., Liu, J., Jin, L., and Zhang, G. (2023). Intersection-based unicast routing using ant colony optimization in software-defined vehicular networks. *Electronics*, 12(7):1620.