

# Offloading de Tarefas de UAVs para Veículos: Uma Análise dos Padrões de Mobilidade em Monitoramento de Regiões

Mateus P. Santos<sup>1</sup>, Paulo A. L. Rego<sup>1</sup>,  
Paulo H. G. Rocha<sup>1</sup>, Alisson B. Souza<sup>1</sup> e José N. Souza<sup>1</sup>

<sup>1</sup>Universidade Federal do Ceará - CE - Brasil

mateussantos14@alu.ufc.br, paulo@dc.ufc.br

paulorocha@great.ufc.br, alisson@ufc.br, neuman@ufc.br

**Abstract.** This study analyzes the efficiency of offloading tasks from Unmanned Aerial Vehicles (UAVs) to ground vehicles in monitoring missions, considering static, circular, and square mobility patterns. A tool was developed to generate UAV traffic scenarios, and an offloading algorithm was adapted. Results show that, despite up to a 7% nominal variation in failure rates for circular mobility, no statistically significant differences are observed among the patterns, nor a relevant impact on offloading time, especially for larger tasks. The findings highlight the potential to optimize UAV mobility for greater computational efficiency, with applications in smart cities, including traffic monitoring, accident detection, and stolen-vehicle tracking, reinforcing the need for further research.

**Resumo.** Este estudo analisa a eficiência do offloading de tarefas de Veículos Aéreos Não Tripulados (UAVs) para veículos terrestres em missões de monitoramento, considerando os padrões de mobilidade estático, circular e quadrangular. Foi desenvolvida uma ferramenta para gerar cenários de tráfego com UAVs, e um algoritmo de offloading foi adaptado. Os resultados mostram que, apesar de uma variação nominal de até 7% nas falhas para mobilidade circular, não se observa diferença estatisticamente significativa entre os padrões, nem impacto relevante no tempo de offloading, especialmente para tarefas de maior complexidade. Os achados destacam o potencial de otimizar a mobilidade dos UAVs para maior eficiência computacional, com aplicações em cidades inteligentes, incluindo monitoramento de tráfego, identificação de acidentes e rastreamento de veículos roubados, reforçando a necessidade de mais estudos.

## 1. Introdução

Veículos aéreos não tripulados, ou *Unmanned Aerial Vehicles* (UAVs), têm emergido como ferramentas indispensáveis em diversos cenários, como vigilância, monitoramento de tráfego e operações críticas, devido à sua versatilidade e capacidade de coleta de dados em tempo real [Ahmed et al. 2023]. No entanto, os UAVs enfrentam desafios significativos relacionados às suas limitações computacionais e energéticas, que restringem sua capacidade de executar tarefas intensivas, como processamento de imagens e análise de dados embarcados [Baktayan et al. 2024][Rocha et al. 2024]. Para mitigar essas limitações, o *offloading* computacional, que transfere tarefas para dispositivos terrestres ou infraestrutura de borda, tem sido amplamente explorado como solução promissora [Souza et al. 2020a].

A eficiência do *offloading* de tarefas em sistemas com UAVs está diretamente relacionada ao padrão de mobilidade desses dispositivos, impactando conectividade, latência e desempenho geral [Rashed and Soyturk 2017]. Cenários de vigilância em tempo real, como cidades inteligentes e redes de veículos, apresentam desafios devido às cargas de dados variáveis e demandas computacionais intensivas. Estudos recentes analisam UAVs com mobilidade estática ou dinâmica em configurações específicas, como UAVs estacionários em interseções de tráfego [Huang et al. 2024] ou em movimento contínuo em comboios de veículos [Zhao et al. 2024]. Embora relevantes, essas abordagens frequentemente se concentram em cenários teóricos ou limitados, deixando lacunas na compreensão de como diferentes padrões de mobilidade impactam o desempenho do sistema, especialmente em aplicações de vigilância urbana.

O objetivo deste estudo é investigar a eficiência do *offloading* de tarefas de UAVs para veículos terrestres em cenários de vigilância, avaliando como diferentes padrões de mobilidade influenciam o desempenho do sistema. Foram realizados experimentos utilizando UAVs configurados com três padrões de mobilidade (estático, circular e quadrangular) para monitorar vias e realizar o *offloading* de *frames* capturados para leitura de placas de veículos. Os experimentos consideraram variáveis como resolução das imagens capturadas e quantidade de tarefas transferidas.

As principais contribuições deste trabalho incluem:

- Desenvolvimento da ferramenta SuUAV para simular UAVs com diferentes mobilidades no SUMO, integrada ao ns-3.
- Análise do impacto de três padrões de mobilidade na eficiência do *offloading*, considerando diferentes resoluções de imagens e cargas de trabalho.

Os resultados indicam que embora a mobilidade circular dos UAVs apresente uma taxa de sucesso de *offloading* nominalmente inferior, os testes estatísticos revelaram que essa diferença não é significativa. Além disso, a análise dos tempos de execução do *offloading* mostram que, não há evidências suficientes para afirmar diferença significativa no desempenho para diferentes padrões de mobilidade, especialmente para tarefas maiores. Os resultados possuem aplicações em monitoramento de tráfego, contribuindo para a otimização dos sistemas de transporte.

O artigo está organizado assim: a Seção 2 revisa trabalhos relacionados, a Seção 3 modela o sistema e o algoritmo de *offloading*, a Seção 4 apresenta os experimentos e resultados, e a Seção 5 conclui com perspectivas futuras.

## 2. Trabalhos Relacionados

O *offloading* de tarefas em sistemas envolvendo UAVs tem se mostrado uma solução promissora para superar as limitações de recursos computacionais e energéticos desses dispositivos, especialmente em cenários críticos como vigilância, monitoramento de tráfego e resgates pós-desastres [Baktayan et al. 2024]. Entretanto, questões como a mobilidade dos UAVs e a variação de carga ainda carecem de investigações mais detalhadas.

[Zhao et al. 2024], por exemplo, explora o *offloading* de tarefas em sistemas de formação de comboios de veículos auxiliados por UAVs, com ênfase na alocação de recursos e otimização do consumo de energia. Os autores modelaram matematicamente o sistema e propuseram algoritmos de tomada de decisão de *offloading* entre UAVs e

os veículos do comboio. Diferentemente do nosso trabalho, os UAVs estão sempre em movimentação, acompanhando o movimento do comboio.

O trabalho de [Huang et al. 2024] investiga o *caching* e o *offloading* em redes de veículos, focando em minimizar atrasos e otimizar a eficiência energética por meio do uso de aprendizado por reforço federado. Assim como o nosso trabalho, os UAVs são utilizados em cenários de vigilância. No entanto, Huang *et al.* limitam-se a UAVs estacionários (pairando em uma interseção de tráfego), enquanto nosso trabalho considera diferentes padrões de mobilidade, como trajetórias circulares e quadrangulares, que impactam diretamente o desempenho do *offloading*.

Os estudos analisados mostram abordagens complementares ao *offloading* de tarefas em UAVs, variando em termos de aplicação, esquemas de comunicação e métodos de otimização. O diferencial do trabalho proposto está na análise de diferentes padrões de mobilidade do UAV, como trajetórias estáticas, circulares e quadrangulares, em um cenário de vigilância.

### 3. Método

Esta seção descreve os padrões de mobilidade de UAVs e as técnicas de *offloading* de tarefas.

#### 3.1. Ferramenta de Mobilidade de UAVs

Para simular a mobilidade dos UAVs, foi desenvolvida uma ferramenta que integra veículos aéreos não tripulados a cenários de tráfego urbano gerados no SUMO. O processo iniciou com a modelagem do cenário através da importação de um mapa real de Manhattan ( $2 \text{ km}^2$ ) do OpenStreetMap em formato .osm, que foi convertido em arquivos .xml no SUMO para simulação do tráfego urbano. Em seguida, um script em Python foi implementado para configurar os parâmetros dos UAVs, incluindo quantidade, altura de operação, posições iniciais, padrão de mobilidade e velocidade de deslocamento. Por fim, o cenário completo foi exportado em formato .tcl para garantir compatibilidade com o ns-3.

Utilizando como base os padrões de mobilidade definidos em [Rashed and Soyturk 2017], que define modelos de mobilidade para UAVs em cenários de aquisição de imagens, foram definidos os três padrões de mobilidade e seus parâmetros:

- **Estático:** UAV fixo em  $(x_0, y_0)$ .
- **Circular:** Movimento em circunferência com centro  $(x_c, y_c)$ , raio  $r$ , velocidade  $v$  e velocidade angular  $\omega$ .
- **Quadrangular:** Trajetória em quadrado com centro  $(x_c, y_c)$ , tamanho do lado  $L$ , velocidade  $v$  e ângulo inicial  $\theta_0$ .

#### 3.2. Offloading

No cenário trabalhado, o UAV atua como cliente do *offloading*, enquanto os veículos terrestres atuam como servidores com capacidades computacionais variáveis. O cliente se comunica com os servidores diretamente através do protocolo IEEE 802.11p. As tarefas envolvem o processamento de imagens capturadas pelo UAV para detectar veículos e realizar a leitura de placas de veículos utilizando *Optical Character Recognition* (OCR). O

algoritmo *Multi-Decision based Offloading* (MDO), proposto em [Rahman et al. 2020], foi adaptado neste trabalho, denominado MDO<sub>A</sub>, para priorizar veículos com maior capacidade de processamento e estimativa de tempo de enlace necessário para a comunicação. A primeira etapa do algoritmo envolve descobrir quais são os veículos próximos ao UAV e quais as capacidades de processamento e tamanho de fila destes. Na sequência, o algoritmo filtra apenas os veículos com estimativa de tempo de enlace maior do que o tempo necessário para fazer *offloading* e seleciona os veículos com maior poder de processamento para executar as tarefas. Utilizou-se a modelagem definida em [Rocha et al. 2022] para a estimativa do tempo de enlace.

Além do algoritmo MDO<sub>A</sub>, foram utilizados outros três algoritmos de *offloading*, com o objetivo de diversificar cenários e analisar o impacto do algoritmo utilizado.

Essas abordagens definem a seleção dos veículos que atuam como servidores no processo de *offloading* de tarefas:

- Servidores Mais Próximos (MP): Este algoritmo prioriza os veículos que estão mais próximos do UAV-cliente, visando minimizar a latência de comunicação e otimizar a transferência de dados, conforme descrito em [Souza et al. 2019];
- Servidores com Maior Tempo de Enlace (MT): Este algoritmo prioriza os veículos que oferecem o maior tempo estimado de enlace para o UAV. A ideia é escolher veículos cujas conexões apresentem maior estabilidade e durabilidade;
- Aleatório (AL): Nesse caso, após a etapa de descoberta, o algoritmo seleciona os veículos de forma aleatória para o *offloading*.

## 4. Experimentos

Esta seção detalha como os cenários de experimentação foram criados, bem como os parâmetros e algoritmos utilizados. Por fim, os resultados são discutidos.

### 4.1. Planejamento

Após a geração dos cenários de mobilidade de veículos e UAVs, experimentos foram conduzidos utilizando a versão 3.29 do ns-3, executados em uma máquina virtual Ubuntu 20.04 LTS com o processador Intel Core i5-10400F e 16 GB de RAM. A Tabela 1 apresenta os principais parâmetros de simulação utilizados.

Nos experimentos, o UAV foi fixado a uma altura de 50 metros e configurado com uma capacidade de processamento limitada (0,25 GHz), devido a restrições energéticas, já os veículos foram definidos com um processamento entre 1.0 e 4.0 GHz, atendendo requisições para execução de tarefas do UAV. O UAV pode fazer *offloading* de 1 a 4 imagens simultaneamente, com tamanhos variando entre 558 kB e 1200 kB, conforme os parâmetros definidos em [Souza et al. 2020b].

Para o padrão estático, o UAV foi inserido no centro do cenário. Para o padrão circular, o centro da circunferência foi definido como o centro do cenário, o raio foi de 40m e a velocidade do UAV foi de 10m/s. E para o padrão quadrangular, foi definido o centro do quadrado como o centro do cenário, lados de 80m e velocidade de 10m/s com ângulo inicial  $\theta_0 = 90^\circ$ .

Para cada cenário, foram executadas 50 simulações, variando o momento em que o UAV inicia o processo de descoberta e decisão de *offloading*. Essa variação teve o

**Tabela 1. Parâmetros da Simulação**

| Parâmetro                                  | Valor                          |
|--|--------------------------------|
| Densidade de veículos (/ km <sup>2</sup> ) | 136                            |
| Densidade de UAV (/ km <sup>2</sup> )      | 1                              |
| Altura do UAV ( m )                        | 50                             |
| Número de configurações                    | 50                             |
| Capacidade da CPU do UAV ( $C$ )           | 0.25 GHz                       |
| Capacidade da CPU dos veículos ( $C$ )     | [1.0, 2.0, 3.0, 4.0] GHz       |
| Uso de CPU da tarefa ( $c$ )               | 3.5 Gigacycles                 |
| Tamanho da tarefa ( $s^{\text{off}}$ )     | [558, 1200] kB                 |
| Número de tarefas ( $s^{\text{off}}$ )     | 1 tarefa, 2 tarefas, 4 tarefas |
| Tamanho do resultado( $s^{\text{down}}$ )  | Dados da placa (1 kB)          |
| Alcance da comunicação                     | 250 m (WAVE)                   |
| Taxa de transferência de dados             | 27 Mbps (WAVE)                 |

objetivo de modificar a disponibilidade de veículos para receber as tarefas. Os resultados obtidos foram então agrupados para o cálculo das métricas de interesse desta pesquisa: a média da taxa de sucesso de *offloading* e a média do tempo total de *offloading*.

## 4.2. Resultados

Esta Seção discute os resultados obtidos para as métricas de interesse dos experimentos, como taxa de sucesso de *offloading* e tempo de execução do *offloading*. A Tabela 2 apresenta os resultados da taxa de sucesso de *offloading* para os dois algoritmos com melhor desempenho ( $\text{MDO}_A$  e MT), considerando diferentes cenários e cargas de trabalho.

Os resultados revelam que, em todos os cenários, a taxa de sucesso diminui à medida que o número e o tamanho das tarefas aumentam. No entanto, essa taxa permanece relativamente alta, especialmente para o algoritmo  $\text{MDO}_A$ . Em contrapartida, o algoritmo MT apresenta taxas de sucesso ligeiramente inferiores, principalmente quando as tarefas são maiores. No cenário com mobilidade quadrangular, o MT supera o  $\text{MDO}_A$ , mas essa superioridade ocorre apenas quando o *offloading* envolve uma única tarefa.

Adicionalmente, observa-se que os cenários estático e quadrangular possuem de taxa de sucesso média maior em relação ao cenário circular. Em comparação com os cenários estático e quadrangular, o cenário circular apresenta uma taxa de falhas nominal até 7% mais alta.

Para confirmar se há uma diferença significativa na taxa de sucesso, foi utilizado o teste ANOVA e não foram encontradas evidências de que o tipo de movimento impacta no desempenho do *offloading*, uma vez que apresentaram estatisticamente o mesmo desempenho.

### 4.2.1. Tempo de Execução de Offloading

A Figura 1 apresenta seis gráficos que mostram os tempos médios de execução do *offloading* com intervalos de confiança de 95%. Em vermelho, também são apresentados os tempos de execução das tarefas quando executadas localmente no UAV. Os gráficos revelam tendências importantes sobre o desempenho dos diferentes algoritmos de *offloading* em três cenários de mobilidade distintos.

**Tabela 2. Taxa de sucesso de offloading (%)**

| Cenário              | Estático         |      | Circular         |       | Quadrangular     |       |
|----------------------|------------------|------|------------------|-------|------------------|-------|
|                      | MDO <sub>A</sub> | MT   | MDO <sub>A</sub> | MT    | MDO <sub>A</sub> | MT    |
| 1 Tarefa de 558 kB   | 100,0            | 98,0 | 94,0             | 97,96 | 98,0             | 100,0 |
| 2 Tarefas de 558 kB  | 98,0             | 96,0 | 90,82            | 97,96 | 98,0             | 98,0  |
| 4 Tarefas de 558 kB  | 94,5             | 88,0 | 85,71            | 93,88 | 92,5             | 90,0  |
| 1 Tarefa de 1200 kB  | 98,0             | 98,0 | 93,88            | 97,96 | 98,0             | 100,0 |
| 2 Tarefas de 1200 kB | 96,0             | 92,0 | 90,82            | 95,92 | 96,0             | 98,0  |
| 4 Tarefas de 1200 kB | 92,0             | 78,0 | 84,69            | 81,63 | 91,5             | 88,0  |

O algoritmo MDO<sub>A</sub> se destaca por apresentar os menores tempos de *offloading* na maioria dos casos, especialmente para tarefas de 558 kB. Isso indica que o MDO<sub>A</sub> é eficiente em termos de tempo, independentemente do cenário ou do número de tarefas. Observa-se que, ao utilizar o MDO<sub>A</sub>, houve uma redução de até 7,5 vezes no tempo de execução das tarefas comparado à execução local (cenário com 4 tarefas de 558 kB). Além disso, é importante notar que, à medida que o tamanho das tarefas aumenta, os tempos de *offloading* também aumentam, embora de forma controlada e proporcional ao tamanho da tarefa.

De maneira geral, os gráficos indicam que realizar *offloading* é vantajoso em todos os cenários analisados. Contudo, vale destacar que o poder de processamento do UAV foi configurado como 1/4 do poder de processamento do veículo mais fraco, o que pode ter favorecido a escolha do *offloading*. Em cenários com UAVs com maior poder computacional, o uso de algoritmos de *offloading* mais avançados deve ser considerado. Nesse contexto, o algoritmo MDO<sub>A</sub> se destaca, sendo o mais eficiente em todos os cenários de mobilidade e para diferentes quantidades de tarefas.

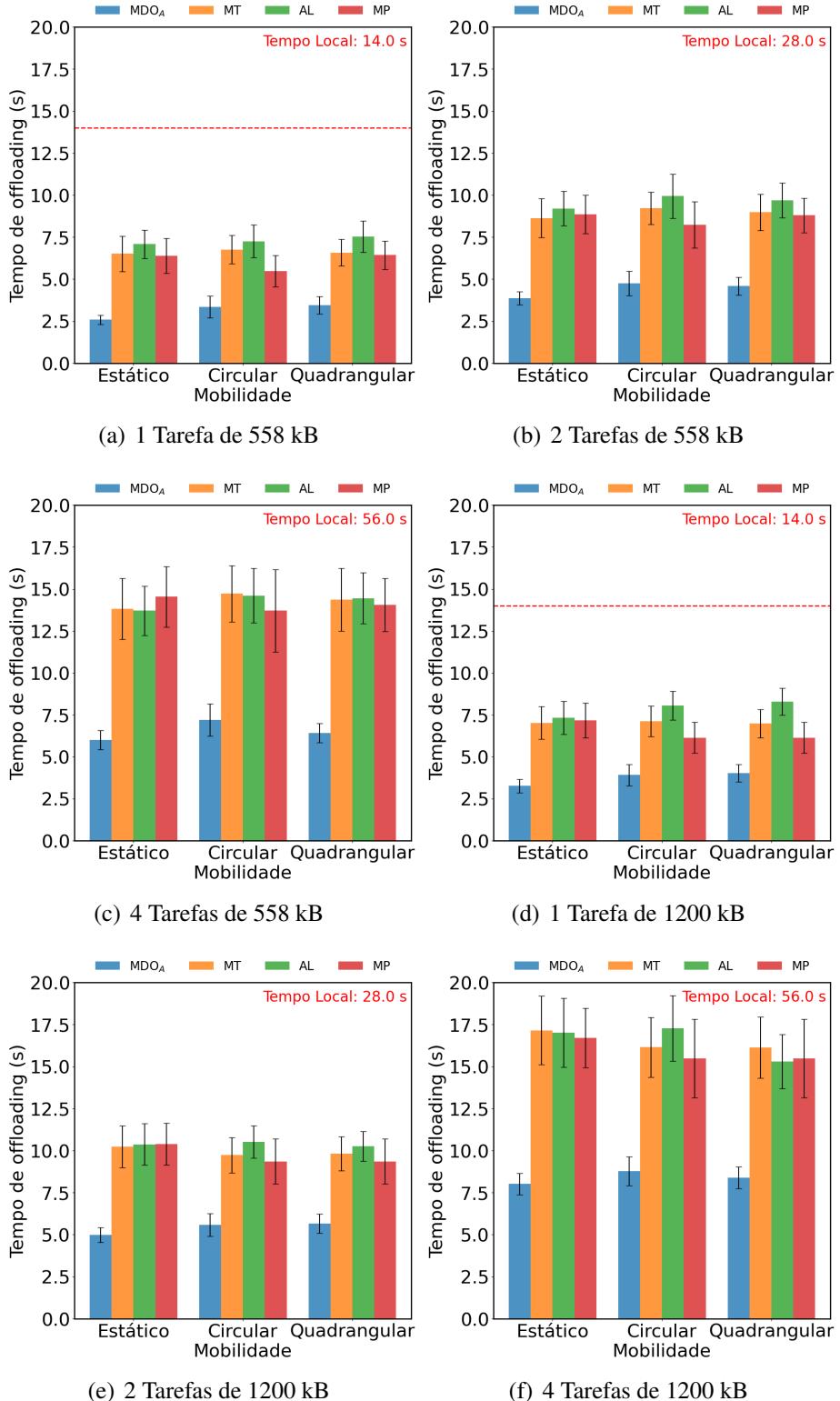
A análise dos intervalos de confiança mostra que os algoritmos MP, MT e AL apresentam desempenhos equivalentes na maioria dos cenários, o que foi corroborado por meio de testes de hipótese (mas por questões de espaço, não são apresentados aqui).

## 5. Conclusão e Trabalhos Futuros

Este trabalho investigou a eficiência do *offloading* de tarefas de UAVs para veículos terrestres em missões de monitoramento de cenários urbanos, com foco na análise dos padrões de mobilidade dos UAVs. A motivação surgiu da necessidade de superar as limitações computacionais e energéticas dos UAVs, utilizando o *offloading* como uma estratégia para melhorar a eficiência. Nesse contexto, o objetivo foi compreender como diferentes padrões de mobilidade influenciam a eficácia do *offloading*, considerando aspectos como taxa de sucesso e tempo de execução.

Os resultados incluem o desenvolvimento de uma ferramenta para inclusão de UAVs com diferentes padrões de mobilidade em simulações e mostraram que, embora a mobilidade circular apresentasse uma taxa de sucesso de *offloading* nominalmente inferior, essa diferença se mostrou estatisticamente irrelevante conforme o teste ANOVA aplicado. Além disso, não houve diferenças significativas nos tempos de execução, especialmente em tarefas maiores, indicando que a mobilidade circular não compromete o desempenho do sistema em termos de tempo.

Trabalhos futuros devem explorar padrões de mobilidade dinâmicos mais comple-



**Figura 1. Tempos de offloading para tarefas de 558 kB e 1200 kB**

xos e desenvolver algoritmos avançados de *offloading* para cenários com UAVs de maior poder computacional. Além disso, é possível realizar uma análise aprofundada de gastos energéticos de UAVs e utilizar modelagens mais complexas levando em conta obstáculos

e interferências. A aplicação em cidades inteligentes, focada na detecção de acidentes e veículos roubados, ainda oferece várias oportunidades de otimização.

## Referências

- [Ahmed et al. 2023] Ahmed, M., Mirza, M. A., Raza, S., Ahmad, H., Xu, F., Khan, W. U., Lin, Q., and Han, Z. (2023). Vehicular communication network enabled cav data offloading: A review. *IEEE Transactions on Intelligent Transportation Systems*, 24(8):7869–7897.
- [Baktayan et al. 2024] Baktayan, A. A., Zahary, A. T., and Al-Baltah, I. A. (2024). A systematic mapping study of uav-enabled mobile edge computing for task offloading. *IEEE Access*.
- [Huang et al. 2024] Huang, J., Zhang, M., Wan, J., Chen, Y., and Zhang, N. (2024). Joint data caching and computation offloading in uav-assisted internet of vehicles via federated deep reinforcement learning. *IEEE Transactions on Vehicular Technology*.
- [Rahman et al. 2020] Rahman, A. U., Malik, A. W., Sati, V., Chopra, A., and Ravana, S. D. (2020). Context-aware opportunistic computing in vehicle-to-vehicle networks. *Vehicular Communications*, 23:100214.
- [Rashed and Soyturk 2017] Rashed, S. and Soyturk, M. (2017). Analyzing the effects of uav mobility patterns on data collection in wireless sensor networks. *Sensors*, 17(2).
- [Rocha et al. 2024] Rocha, C., Pacheco, L., Bastos, L., Rosário, D., and Cerqueira, E. (2024). Coop: Um algoritmo de computação e offloading para redes terrestres assistidas por vant. In *Anais Estendidos do XLII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 193–200, Porto Alegre, RS, Brasil. SBC.
- [Rocha et al. 2022] Rocha, P., Souza, A., Maia, G., Mattos, C., Silva, F. A., Rego, P., Nguyen, T. A., and Lee, J.-W. (2022). Evaluating link lifetime prediction to support computational offloading decision in vanets. *Sensors*, 22(16):6038.
- [Souza et al. 2019] Souza, A. B., Leal Rego, P. A., and de Souza, J. N. (2019). Exploring computation offloading in vehicular clouds. In *2019 IEEE 8th International Conference on Cloud Networking (CloudNet)*, pages 1–4.
- [Souza et al. 2020a] Souza, A. B., Rego, P. A., Carneiro, T., Rodrigues, J. D. C., Reboucas Filho, P. P., Souza, J. N., Chamola, V., De Albuquerque, V. H. C., and Sikdar, B. (2020a). Computation offloading for vehicular environments: A survey. *IEEE Access*, 8:198214–198243.
- [Souza et al. 2020b] Souza, A. B. d., Rego, P. A. L., Rocha, P. H. G., Carneiro, T., and Souza, J. N. d. (2020b). A task offloading scheme for wave vehicular clouds and 5g mobile edge computing. In *2020 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE.
- [Zhao et al. 2024] Zhao, P., Kuang, Z., Guo, Y., and Hou, F. (2024). Task offloading and resource allocation in uav-assisted vehicle platoon system. *IEEE Transactions on Vehicular Technology*.