

# VORTEX: Descarregamento de Dados Sensível à Latência ao Longo do Contínuo Edge–Cloud para Veículos Autônomos Conectados

Mariese C A dos Santos<sup>1</sup>, João P. M. Fernandes<sup>1</sup>, Thiago L. M. Mello<sup>1</sup>,  
António Grilo<sup>2</sup>, Maycon L. M. Peixoto<sup>1</sup>

<sup>1</sup>Instituto de Computação – Universidade Federal da Bahia (UFBA)

{marieseconceicao@ufba.br, joaopmpf@ufba.br, mayconleo@gmail.com},

<sup>2</sup>Instituto Superior Técnico – Universidade de Lisboa (INESC-ID/IST/ULisboa)

{antonio.grilo@inov.pt}

**Abstract.** *Connected Autonomous Vehicles (CAVs) generate massive sensory data streams that must be processed under strict latency constraints, often exceeding onboard computational and storage capabilities. This paper presents VORTEX, an object-level offloading framework that prioritizes detected objects rather than raw video streams to improve deadline compliance and bandwidth efficiency across the edge–cloud continuum in heterogeneous LTE/5G vehicular networks. VORTEX employs a distributed, deadline-aware heuristic that orchestrates object placement along the continuum, dynamically leveraging edge proximity while exploiting cloud elasticity when deadlines become infeasible. Results showed that VORTEX achieves 99.1% deadline satisfaction, reduces cloud forwarding, and delivers statistically significant latency improvements compared to EDF, LLF, and SDF baselines.*

**Resumo.** *Veículos Autônomos Conectados (CAVs - Connected Autonomous Vehicles) geram fluxos massivos de dados sensoriais que precisam ser processados sob rigorosas restrições de latência, frequentemente excedendo as capacidades computacionais e de armazenamento embarcadas. Este artigo apresenta o VORTEX, um framework de descarregamento em nível de objeto que prioriza objetos detectados em vez de fluxos de vídeo brutos para melhorar o cumprimento dos prazos e a eficiência do uso de largura de banda ao longo do contínuo edge–cloud em redes veiculares heterogêneas LTE/5G. O VORTEX emprega uma heurística distribuída e sensível a prazos que orchestra o posicionamento dos objetos ao longo do contínuo, explorando dinamicamente a proximidade da borda e a elasticidade da nuvem quando os prazos se tornam inviáveis. Os resultados mostram que o VORTEX alcança 99,1% de cumprimento de prazos, reduz o encaminhamento para a nuvem e apresenta melhorias de latência estatisticamente significativas em comparação com os algoritmos de referência EDF, LLF e SDF.*

## 1. Introdução

O avanço do nível de automação em Veículos Autônomos Conectados (CAVs – Connected Autonomous Vehicles) tem impulsionado a geração de volumes massivos de dados

sensoriais por dispositivos embarcados, especialmente câmeras e sistemas de Detecção e Telemetria de Luz (LiDAR - Light Detection and Ranging) [Saleem et al. 2021, Ferreira et al. 2025].

Embora essenciais para o mapeamento do ambiente, o processamento desses fluxos em tempo real frequentemente excede as capacidades computacionais a bordo, criando gargalos que podem comprometer a responsividade do sistema [Freire et al. 2025, Ferreira et al. 2025, Rajareddy et al. 2025a, Peixoto 2024]. Nesse cenário, mecanismos eficientes de descarregamento tornam-se fundamentais, não só para mitigar limitações computacionais, mas para viabilizar a percepção cooperativa, permitindo que a infraestrutura compartilhe objetos detectados com veículos próximos que possuam obstruções em sua linha de visão. A efetividade dessa colaboração depende de baixa latência, alta confiabilidade e da disponibilidade de infraestrutura computacional próxima aos veículos [Nguyen et al. 2024].

A computação de borda reduz o atraso fim a fim ao aproximar recursos computacionais da origem dos dados, mas sua implantação limitada e restrições de capacidade dificultam o atendimento consistente às demandas veiculares [Maia et al. 2025, Peixoto 2024]. Em contraste, a nuvem oferece elasticidade praticamente ilimitada, ao custo de latências incompatíveis com aplicações sensíveis ao tempo. Arquiteturas híbridas buscam equilibrar essas propriedades [Peixoto 2024, Silva et al. 2023], porém tratar borda e nuvem como camadas discretas restringe a tomada de decisão em ambientes altamente dinâmicos. Esse contexto tem motivado a adoção do uso contínuo borda–nuvem como um modelo capaz de orquestrar recursos distribuídos de forma flexível e sensível ao contexto.

Apesar desses avanços, uma limitação persistente do estado da arte reside na granularidade das decisões de descarregamento. A maioria das abordagens concentra-se na redução da carga computacional, mas negligencia a otimização dos fluxos associados à detecção de objetos [Qiu et al. 2024]. A transmissão de fluxos de vídeo completos eleva significativamente o consumo de largura de banda e a latência de processamento [Maresch and Nastic 2024], agravando o uso de recursos já escassos na borda e dificultando o cumprimento rigorosos dos prazos. Além disso, decisões de descarregamento frequentemente consideram parâmetros isolados, como o prazo ou tamanho dos dados, sem capturar adequadamente as interdependências entre urgência temporal, custo de transmissão e viabilidade de processamento ao longo do contínuo computacional.

Para enfrentar essas limitações, este trabalho propõe o **VORTEX**, uma estratégia de descarregamento de dados em nível de objeto projetada para operar ao longo do contínuo borda–nuvem em redes veiculares complexas e dinâmicas. Diferentemente das abordagens que transmitem fluxos de vídeo completos, o VORTEX prioriza objetos detectados por meio de um processo de seleção sensível a prazos e de alocação adaptativa, reduzindo o consumo de largura de banda enquanto minimiza a latência. A estratégia orquestra dinamicamente o posicionamento dos dados ao longo do contínuo, explorando a proximidade da borda para objetos com restrição do tempo de vida viáveis e recorrendo à elasticidade da nuvem quando tais restrições não podem ser atendidas localmente, favorecendo maior cumprimento de prazos e tornando o descarregamento a base fundamental para a percepção cooperativa, permitindo que a borda consolide e compartilhe detecções críticas com veículos que possuem obstruções em sua linha de visão direta, garantindo o

cumprimento de prazos.

As principais contribuições deste artigo são: (i) um mecanismo de descarregamento em nível de objeto que substitui a transmissão de fluxos de vídeo completos, otimizando a comunicação CAV–Borda–Nuvem e reduzindo o consumo de largura de banda; (ii) uma estratégia sensível a prazos que maximiza o número de objetos processados sob restrições de tempo real; e (iii) um framework adaptativo de integração CAV–Borda–Nuvem que orquestra o posicionamento dos objetos ao longo do contínuo borda–nuvem, direcionando objetos com prazos viáveis para a borda e encaminhando os demais para a nuvem, de modo a otimizar recursos e preservar a baixa latência.

## 2. Trabalhos Relacionados

O crescimento exponencial no número de sensores integrados em CAVs levou a um aumento substancial nos dados gerados, tornando o uso da computação de borda essencial para garantir baixa latência. Contribuições incluem a abordagem de Silva et al. [Lee et al. 2015], que utiliza recursos de borda acoplados a Unidades de Beira de Estrada (RSUs - Road Side Units) via DSRC, e o método de Costa et al. [Raja et al. 2020], que aproveita estações base operando em redes 4G. Embora eficazes em ambientes controlados, ambas as abordagens são limitadas por sua dependência de infraestruturas de rede específicas.

A maioria dos trabalhos existentes concentra-se na seleção da localização ideal para o descarregamento de dados, muitas vezes negligenciando a decisão da seleção dos dados dentro do CAV. Os critérios comuns utilizam o tamanho do dado ou o prazo. No entanto, considerar somente o tamanho resulta em violações de prazos, enquanto priorizar apenas o prazo pode elevar a latência. Essas compensações destacam a necessidade de estratégias que considerem múltiplos parâmetros. Oliveira et al. [Oza et al. 2024] explora a duração da parada de um veículo em semáforos para descarregar dados com prazos iminentes. De forma semelhante, Pereira et al. [Nguyen et al. 2024] introduzem uma abordagem em que os dados com o prazo mais curto são priorizados para descarregamento na borda.

Estratégias recentes de descarregamento veicular priorizam variáveis isoladas ou combinadas, como tamanho, prazo e custo computacional [Meng et al. 2019, Saleem et al. 2021, Kumar et al. 2024, Rajareddy et al. 2025b]. No entanto, essas abordagens apresentam limitações ao não explorarem ambientes de redes heterogêneas ou a integração eficiente entre borda e nuvem considerando simultaneamente prioridade, prazo e tamanho (ver Tabela 1).

Em contraste a esses trabalhos, o VORTEX implementa um algoritmo de seleção que incorpora simultaneamente prazo, tamanho e prioridade dos dados, permitindo o acesso a múltiplos tipos de rede, recursos de computação de borda e nuvem. Essa abordagem preenche a lacuna existente em estudos anteriores ao unir os componentes separados de configuração de rede e seleção de dados em um único mecanismo.

## 3. Modelo do Sistema e Formulação do Problema

Este trabalho considera uma arquitetura veicular hierárquica que integra estações rádio-base, computação de borda e nuvem ao longo de um contínuo de descarregamento. As

**Tabela 1. Comparação VORTEX e Trabalhos Relacionados: (1) Redes Heterogêneas, (2) Computação na Borda, (3) Computação na Nuvem, (4) Seleção por Prazo, (5) Seleção por Tamanho, (6) Priorização de Dados.**

Trabalhos de Referência	Características					
	1	2	3	4	5	6
Dove [Lee et al. 2015]	×	✓	×	×	×	×
PrioritySched [Taherkhani and Pierre 2016]	×	×	×	×	✓	✓
DEDAS [Meng et al. 2019]	×	×	✓	✓	✓	✓
I-ANDSF/IR-DON [Raja et al. 2020]	×	✓	×	×	×	×
BandwidthSched [Agarwal et al. 2020]	×	✓	✓	✓	×	×
DIVINE[Saleem et al. 2021]	×	✓	×	×	✓	✓
AdaptiveDeadline [Kumar et al. 2024]	×	×	✓	✓	✓	✓
DeadlineOffVEC [Oza et al. 2024]	×	×	×	✓	×	×
DeadlineOffEdge [Nguyen et al. 2024]	×	✓	✓	✓	×	×
MobilitySched [Rajareddy et al. 2025b]	×	×	✓	✓	✓	✓
<b>VORTEX (Este Trabalho)</b>	✓	✓	✓	✓	✓	✓

estações conectam CAVs a nós de borda colocalizados, enquanto a nuvem fornece capacidade virtualmente ilimitada para dados não sensíveis à latência.

A rede é modelada como um grafo direcionado  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , em que  $\mathcal{V}$  compreende veículos  $\{v_1, \dots, v_M\}$  e estações rádio-base  $\{b_1, \dots, b_J\}$ . Cada aresta  $e_{m,j} \in \mathcal{E}$  representa um enlace sem fio associado a um peso  $w_{m,j}$  que captura a qualidade do canal. Cada veículo  $v_m$  gera fluxos de vídeo  $F_m = \{f_{m,1}, \dots, f_{m,I}\}$ , onde cada vídeo  $f_{m,i} = \{\theta_{m,i}, \sigma_{m,i}, \Omega_{m,i}\}$  é definido por tempo de captura, tamanho e conjunto de objetos detectados. Cada objeto  $\omega_{m,k} \in \Omega_{m,i}$  é descrito por  $\omega_{m,k} = [\theta_{m,k}, \psi_{m,k}, \varphi_{m,k}, \lambda_{m,k}, \sigma_{m,k}]$ , correspondendo ao tempo de detecção, posição, tipo, tempo de vida ou prazo e tamanho. Assume-se que  $\lambda_{m,k}$  é inversamente proporcional à distância ao veículo, objetos mais próximos representam um risco de colisão, o que reduz o tempo disponível para processamento e reação do sistema. E  $\sum_{\omega_{m,k} \in \Omega_{m,i}} \sigma_{m,k} \leq \sigma_{m,i}$ .

Veículos possuem capacidade finita de armazenamento  $C_m^{CAV}$ , enquanto cada nó de borda dispõe de capacidade  $C_j^{BS}$ . No instante  $t$ , as capacidades disponíveis são  $C_{m,t}^{CAV} = C_m^{CAV} - Q_{m,t}^{CAV}$  e  $C_{j,t}^{BS} = C_j^{BS} - Q_{j,t}^{BS}$ . Um objeto pode ser mantido localmente se  $\sigma_{m,k} \leq C_{m,t}^{CAV}$  e descarregado quando satisfaz simultaneamente  $\lambda_{m,k} > t$ , ou seja, no instante  $t$  o tempo de vida do objeto ainda é válido, e  $\sigma_{m,k} \leq C_{j,t}^{BS}$ .

A urgência do descarregamento é capturada pelo tempo de transmissão e processamento  $t_p(\omega_{m,k}) = \sigma_{m,k} / \kappa_{m,t}$ , onde  $\kappa_{m,t}$  denota a taxa de uplink. O tempo de colaboração restante é definido como  $t_c(\omega_{m,k}) = \lambda_{m,k} - t - t_p(\omega_{m,k})$ , significando o restante do tempo de vida após o descarregamento no instante  $t$ . Objetos com  $t_c(\omega_{m,k}) \leq 0$ , que extrapolaram o tempo de vida, são encaminhados à nuvem após o descarregamento dos objetos viáveis na borda. A priorização baseia-se no número potencial de transmissões  $NT = t_c(\omega_{m,k}) / t_p(\omega_{m,k})$ : objetos com  $NT \leq 1$  seguem Menor Tamanho Primeiro (EDF - Earliest Deadline First), enquanto os demais são ordenados por tamanho via Menor Tamanho Primeiro (SDF - Shortest Data First).

A latência fim a fim para descarregar um objeto ao nó de borda  $b_j$  é modelada por  $\mathcal{L}_{m,j,k} = d_{\text{resource}} + d_{\text{queue}} + d_{\text{tx}}$ , onde  $d_{\text{resource}}$  denota o atraso da disponibilidade do

recurso da estação base  $b_j$  para o veículo  $v_m$  descarregar o objeto  $\omega_{m,k}$ ,  $d_{\text{queue}}$  é o atraso que o objeto  $\omega_{m,k}$  enfrenta na fila de transmissão para o nó edge em  $b_j$ , e  $d_{\text{tx},m,k}$  é o atraso na transmissão do objeto. Assim, a decisão de descarregamento deve equilibrar restrições de prazo, capacidade e latência ao longo do contínuo computacional. Em cada instante  $t$ , variáveis binárias  $x_{t,m,j,k} \in \{0, 1\}$  indicam o envio do objeto  $\omega_{m,k}$  para  $b_j$ , buscando maximizar o número de objetos descarregados dentro do prazo e, subsequentemente, o volume total descarregado sem violar restrições de recursos. Essa formulação corresponde a um Problema da Mochila Multidimensional Multiperíodo com Programação Linear Inteira (PMMMP-PLI), sendo NP-difícil devido ao acoplamento temporal, às restrições de capacidade e à natureza binária das decisões.

Assume-se que veículos operam sob cobertura 4G/LTE, 5G/NR ou regiões sobrepostas, selecionando dinamicamente uma estação base conforme as condições do canal. As decisões são independentes entre veículos, enquanto os nós de borda particionam recursos por aplicação e priorizam objetos com prazos válidos.

## 4. VORTEX

Dada a inviabilidade computacional de resolver o PMMMP-PLI de forma ótima em ambientes veiculares altamente dinâmicos, torna-se necessário um mecanismo de decisão capaz de operar sob informação incompleta e restrições temporais rigorosas. Nesse contexto, este trabalho deriva o algoritmo de Troca de Objetos Veiculares em Tempo Real (VORTEX - Vehicular Object Real-Time Exchange), uma heurística online e distribuída projetada para aproximar o objetivo de maximização do número de objetos descarregados dentro de seus respectivos tempos de vida. O VORTEX fundamenta-se em três princípios de projeto: (i) decisões em granularidade fina, realizadas no nível de objeto; (ii) priorização sensível à urgência temporal, reduzindo perdas decorrentes da expiração dos dados; e (iii) adaptação contínua ao estado do sistema, considerando taxa de uplink e capacidade disponível na borda. Ao depender exclusivamente de informações localmente observáveis, a estratégia torna-se viável para implantação prática em cenários veiculares caracterizados por alta mobilidade e variabilidade de rede.

### 4.1. Arquitetura Operacional do VORTEX

O VORTEX opera sobre uma arquitetura hierárquica composta por veículos, estações rádio-base, nós de borda e nuvem, permitindo decisões de descarregamento ao longo do contínuo computacional. Os veículos transmitem objetos detectados para uma estação base selecionada conforme as condições do canal, enquanto cada estação está conectada a um nó de borda responsável pelo armazenamento e gerenciamento inicial dos dados. A nuvem atua como repositório complementar para objetos expirados ou não acomodados na borda.

Ao receber um objeto, o nó de borda verifica a existência de registros semanticamente equivalentes com base no tipo e na proximidade espacial. Quando identificado um registro anterior, este é encaminhado à nuvem e substituído pela observação mais recente, garantindo consistência temporal do cache e evitando a persistência de dados obsoletos. Objetos cujo tempo de vida expira são automaticamente transferidos para a nuvem, preservando recursos de armazenamento na borda.

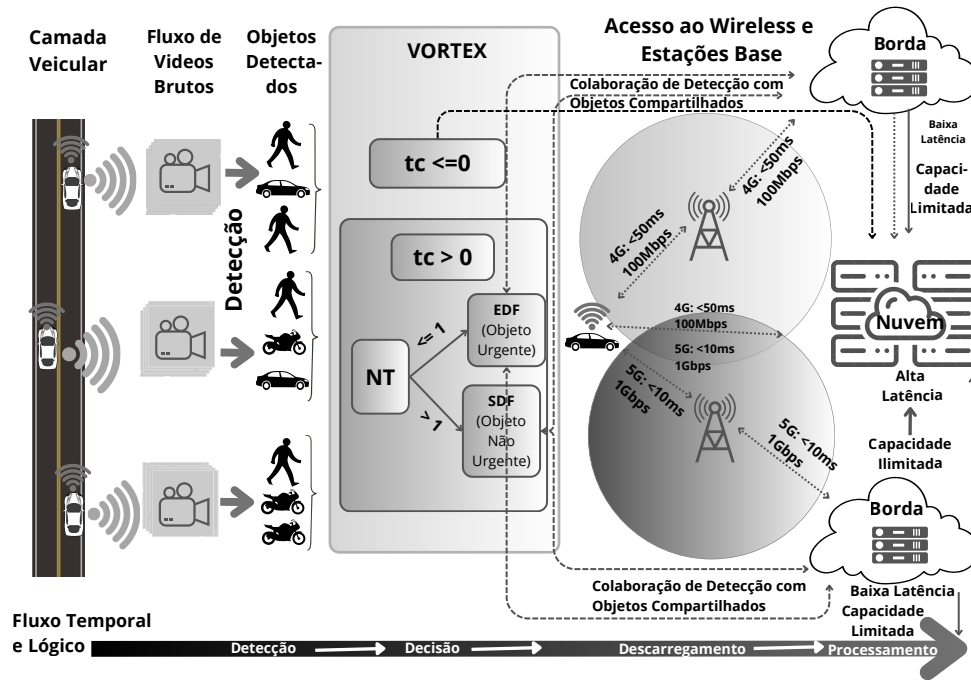


Figura 1. Visão geral do VORTEX: descarregamento em nível de objeto com reconhecimento de prazos ao longo do contínuo borda–nuvem.

## 4.2. Estrutura Algorítmica do VORTEX

O VORTEX decompõe a decisão de descarregamento em três procedimentos distribuídos executados no veículo e no nó de borda: (i) decisão de descarregamento no veículo, (ii) gerenciamento de objetos na borda e (iii) ampliação da percepção cooperativa. Essa decomposição permite decisões oportunísticas e sensíveis ao contexto sem depender de conhecimento global do sistema. Cada decisão possui complexidade polinomial em relação ao número de objetos locais, tornando a estratégia adequada para execução online sob restrições temporais rigorosas. A seguir, detalhamos cada um desses procedimentos.

### 4.2.1. Agendamento e Descarregamento no Veículo

O Algoritmo 1 é executado no veículo  $v_m$  e inicia com o envio de uma solicitação de descarregamento contendo o volume total de dados candidatos. Após receber a capacidade disponível do nó de borda  $C_{j,t}^{BS}$ , o veículo identifica os objetos elegíveis e realiza a seleção iterativa daqueles a serem transmitidos.

Um objeto  $\omega_{m,k}$  é considerado elegível se  $\lambda_{m,k} > t$  e  $\sigma_{m,k} \leq C_{j,t}^{BS}$ . Para cada objeto, calculam-se o tempo de transmissão e processamento  $t_p(\omega_{m,k})$  e o tempo de colaboração restante  $t_c(\omega_{m,k})$ , sendo a priorização baseada no número potencial de transmissões  $NT = t_c/t_p$  após chegar a edge. Objetos com  $NT \leq 1$  são tratados como urgentes e ordenados segundo Menor Prazo Primeiro (EDF - Earliest Deadline First), reduzindo a probabilidade de expiração, enquanto os demais seguem Menor Tamanho Primeiro (SDF - Shortest Data First), favorecendo transmissões rápidas e melhor uso da capacidade. Objetos inviáveis para processamento na borda são encaminhados à nuvem

**Algoritmo 1: Agendamento e Descarregamento no Veículo**


---

```

1 Etapa 1: Solicitação de Descarregamento
2  $v_m$  envia  $\sum_{k=1}^K \sigma_{m,k}$  para  $b_j$ ;
3 Etapa 2: Recebimento da Resposta à Solicitação de Descarregamento e
   Descarregamento dos Objetos Dentro do Prazo
4 enquanto  $(\exists \omega_{m,k}$  em que  $(\lambda_{m,k} > t) \ \& \ (\sigma_{m,k} \leq C_{j,t}^{BS}))$  faça
5   Calcule  $t_p(\omega_{m,k}) = \frac{\sigma_{m,k}}{\kappa_{m,t}}$ ;
6   Calcule  $t_c(\omega_{m,k}) = \lambda_{m,k} - t - t_p(\omega_{m,k})$ ;
7   Calcule  $NT(\omega_{m,k}) = t_c(\omega_{m,k})/t_p(\omega_{m,k})$ ;
8   se  $(t_c(\omega_{m,k}) > 0)$  então
9     se  $(NT(\omega_{m,k}) \leq 1)$  então
10       se  $\lambda_{m,k} = \min(\lambda_{m,1}, \dots, \lambda_{m,K})$  então
11         Envio de  $\omega_{m,k}$  para o nó de borda de  $b_j$ ;
12          $C_{m,t}^{CAV} = C_{m,t}^{CAV} + \sigma_{m,k}$ .
13       senão
14         se  $\sigma_{m,k} = \min(\sigma_{m,1}, \dots, \sigma_{m,K})$  então
15           Envio de  $\omega_{m,k}$  para o nó de borda de  $b_j$ ;
16            $C_{m,t}^{CAV} = C_{m,t}^{CAV} + \sigma_{m,k}$ .
17 Etapa 3: Descarregamento dos Objetos Após o Prazo
18 enquanto  $(\nexists \omega_{m,k}$  com  $t_c(\omega_{m,k}) > 0)$  faça
19   se  $(\lambda_{m,k} = \min(t - \lambda_{m,1}, \dots, t - \lambda_{m,K}))$  então
20     Envio de  $\omega_{m,k}$  para a nuvem;
21      $C_{m,t}^{CAV} = C_{m,t}^{CAV} + \sigma_{m,k}$ .

```

---

após o descarregamento dos viáveis, e a cada transmissão o veículo atualiza sua capacidade disponível, permitindo que decisões subsequentes reflitam o estado corrente do sistema.

#### 4.2.2. Gerenciamento de Objetos na Borda

O Algoritmo 2 dissemina continuamente a capacidade disponível do nó de borda para os veículos ativos e gerencia a recepção dos objetos. Ao detectar redundância semântica e espacial do objeto, o registro mais antigo é migrado para a nuvem e substituído pelo novo, preservando a atualidade das informações armazenadas.

Essa interação em circuito fechado entre veículos e borda evita sobrecarga de armazenamento e permite que as decisões de descarregamento se adaptem dinamicamente à disponibilidade de recursos.

#### 4.2.3. Ampliação da Percepção Cooperativa

O Algoritmo 3 explora o cache da borda para identificar objetos relevantes não detectados localmente por veículos próximos, expandindo seu campo perceptivo e contribuindo para uma visão mais abrangente do ambiente. Objetos cujo tempo de vida expira são automaticamente transferidos para a nuvem, pois não podem colaborar com outros veículos, liberando capacidade computacional para novos objetos.

---

**Algoritmo 2: Gerenciamento de Objetos no Nó de Borda**

---

```

1 Etapa 1: Resposta de  $b_j$  para  $v_m$ 
2 Envio da capacidade de armazenamento disponível  $C_{j,t}^{BS}$  para  $v_m$ .
3 Etapa 2: Recebimento dos Objetos Detectados
4 para cada  $\omega_{j,l}$  armazenado no nó de borda de  $b_j$  faça
5   se  $((\varphi_{m,k} = \varphi_{j,l}) \ \& \ (\psi_{m,k} \approx \psi_{j,l}))$  então
6     Envio de  $\omega_{j,l}$  para a cloud;
7      $C_{j,t}^{BS} = C_{j,t}^{BS} + \sigma_{j,l}$ .
8     Armazenamento de  $\omega_{m,k}$  no nó de borda de  $b_j$ ;
9      $C_{j,t}^{BS} = C_{j,t}^{BS} - \sigma_{m,k}$ .
10 se  $\omega_{m,k}$  não está armazenado no nó de borda de  $b_j$  então
11   Armazenamento de  $\omega_{m,k}$  no nó de borda de  $b_j$ ;
12    $C_{j,t}^{BS} = C_{j,t}^{BS} - \sigma_{m,k}$ .

```

---



---

**Algoritmo 3: Suporte à Percepção Cooperativa na Borda**

---

```

1 Etapa 1: Verificação dos Objetos Não Detectados Pelo Veículo
2 para cada  $\omega_{j,l} \in b_j$  faça
3   se  $((v_{j,l} \neq v_m) \ \& \ (\psi_{j,l} \approx \psi_m))$  então
4     Envio de  $\omega_{j,l}$  para  $v_m$ .
5 Etapa 2: Envio dos Objetos para a Nuvem
6 para cada  $\omega_{j,l} \in b_j$  faça
7   se  $(\lambda_{j,l} \leq t)$  então
8     Envio de  $\omega_{j,l}$  para a Nuvem;
9      $C_{j,t}^{BS} = C_{j,t}^{BS} + \sigma_{j,l}$ ;
10    Envio de  $C_{j,t}^{BS}$  para  $v_m$ .

```

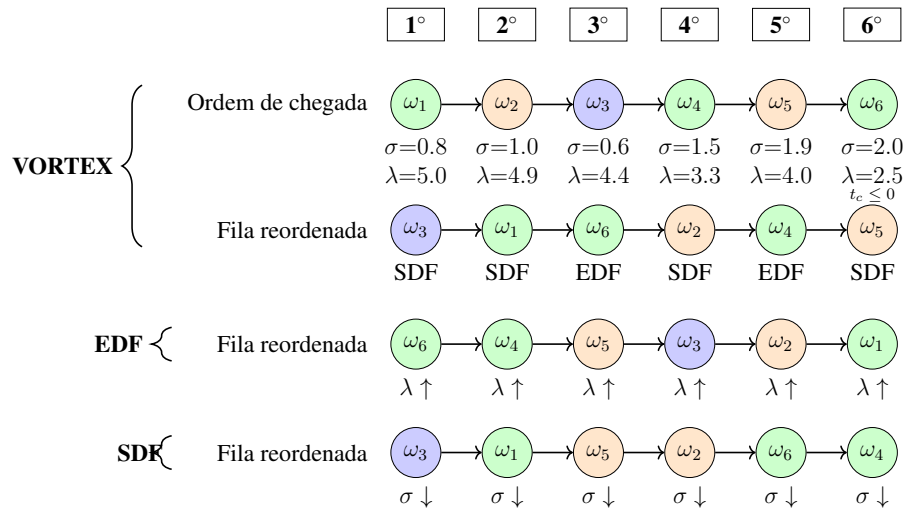
---

Em conjunto, esses procedimentos estabelecem um mecanismo adaptativo capaz de responder à variabilidade do ambiente veicular sem depender de conhecimento global. Como resultado, o VORTEX favorece decisões oportunísticas nas quais objetos com maior relevância temporal são naturalmente priorizados pela dinâmica do sistema.

### 4.3. Exemplo Ilustrativo

A Figura 2 apresenta um exemplo ilustrativo que evidencia como o VORTEX reordena a fila de descarregamento em relação a políticas clássicas de escalonamento. Considera-se um único nó de borda associado à estação base  $b_j$  e um conjunto de seis objetos que chegam sequencialmente ao veículo e se tornam candidatos ao descarregamento. Para parametrização, assume-se um instante de decisão  $t = 0$  e uma taxa de uplink constante  $\kappa_{m,t} = 2.0$  MB/s. Cada objeto é descrito por seu tamanho  $\sigma$  e tempo de vida  $\lambda$ , e o VORTEX computa  $t_c(\omega)$  e  $NT(\omega)$  após cada objeto ser descarregado. Objetos com  $NT \leq 1$  são tratados como urgentes e ordenados por EDF, enquanto os demais são ordenados por SDF, refletindo o compromisso entre urgência temporal e eficiência de transmissão.

Nesse exemplo, o VORTEX a cada instante do tempo reordena a lista após descarregar um objeto. Como a seleção inicia em  $t = 0$ , todos com  $N_t > 1$ , e os objetos  $\omega_3$  e  $\omega_1$  são sequencialmente descarregados utilizando o SDF. Logo após,  $\omega_6$  apresenta  $N_t \leq 1$  e será descarregado pelo EDF. Quando chega o momento de  $\omega_5$  ser descarregado, apresenta



**Figura 2. Exemplo ilustrativo de reordenação da fila de descarregamento para o nó de borda: o VORTEX combina EDF para objetos urgentes ( $NT \leq 1$ ) e SDF para os demais ( $NT > 1$ ), enquanto EDF e SDF reordenam toda a fila por prazo e tamanho, respectivamente.**

$t_c \leq 0$ , e, conseqüentemente, será descarregado na cloud. Assim, o exemplo evidencia que a política híbrida do VORTEX aproxima a decisão online do compromisso requerido pela formulação: maximizar o descarregamento dentro do tempo de vida sob restrições simultâneas de prazo e capacidade.

## 5. Configuração dos Experimentos

Os experimentos foram conduzidos com o objetivo de caracterizar, de forma controlada e reprodutível, o comportamento do VORTEX em cenários veiculares com suporte a comunicações celulares. O ambiente de simulação integra OMNeT++ 5.6.2 como núcleo de eventos discretos, Simu5G 1.1.0 para modelagem da pilha 5G, INET 4.2.2 para os protocolos de rede, Veins 5.2 para acoplamento entre rede e mobilidade, e SUMO 1.8.0 para geração do tráfego veicular. Os principais parâmetros adotados são apresentados na Tabela 2. Os rastros gerados foram posteriormente processados em Python 3 para garantir uniformidade na coleta e análise estatística dos resultados.

O cenário considera um cruzamento urbano de múltiplas vias com fluxo contínuo de veículos conectados e autônomos, no qual decisões de descarregamento devem ser tomadas sob restrições simultâneas de tempo, capacidade de rede e disponibilidade de recursos computacionais. A infraestrutura incorpora suporte a LTE e 5G NR, enlaces de alta capacidade e mecanismos de dual connectivity, possibilitando avaliar o comportamento do sistema diante de heterogeneidade tecnológica e eventos como handovers. A parametrização privilegia valores compatíveis com implantações realistas, buscando equilibrar granularidade de modelagem e viabilidade computacional, ao mesmo tempo em que assegura repetibilidade por meio de múltiplas execuções com sementes independentes.

A avaliação do VORTEX é realizada por meio de comparação direta com políticas clássicas de escalonamento amplamente consolidadas na literatura: Menor Prazo Primeiro (EDF - Earliest Deadline First), Menor Tamanho Primeiro (SDF - Shortest Data First) e

Tabela 2. Configuração do Cenário de Simulação

Parâmetro	Valor	Parâmetro	Valor
<b>Configurações Gerais e Ambiente</b>			
Tempo de Simulação / Repetições	215 s / 10 Seeds	Playground (OMNeT++)	10.000m × 10.000m × 50m
Gerenciador de Mobilidade	VeinsInetMobility (TraCI)	Intervalo de Atualização (Veins)	0,1 s
<b>Configurações de Mobilidade (SUMO - Simulation of Urban MObility)</b>			
Cenário Viário	Cruzamento de 4 vias	Área da Malha Viária	3.000m × 3.000m
Velocidade Máxima	13,8 m/s (50 km/h)	Número de Faixas por Via	2 faixas
Fluxo de Veículos	31 CAVs (1 veh/s)	Controle de Tráfego	Semafórico
<b>Infraestrutura e Núcleo de Rede (Core)</b>			
Núcleo de Rede	UPF / PGW	Interfaces de Enlace	Eth10G (10 Gbps)
Interface de Coordenação	X2 (gNB ↔ eNB)	Dual Connectivity	Habilitado (IDs 1 e 2)
Capacidade de Armazenamento do Veículo	10G	da Borda(4G/5G)	500G/1T
<b>Camada Física e Protocolos (PHY/MAC)</b>			
Bandas LTE	100 RBs (20 MHz)	Bandas 5G NR	270 RBs (50 MHz)
Potência de Transmissão	46 dBm	Latência de Handover	50 ms
Tipos de NIC	LteNicEnbD2D / NRNicEnb	Intervalo de Broadcast	1 s
<b>Camada de Aplicação</b>			
Tamanho do Vídeo (TV)	$U(0.4MiB; 3.0Mib)$	Tamanho do Objeto (TO)	$U(0.4MiB; (TV - soma(TO)))$
Tamanho do Pacote	600.000 Bytes		
Intervalo de Envio	20 ms	Portas de Comunicação	1000 / 3000 / 5000

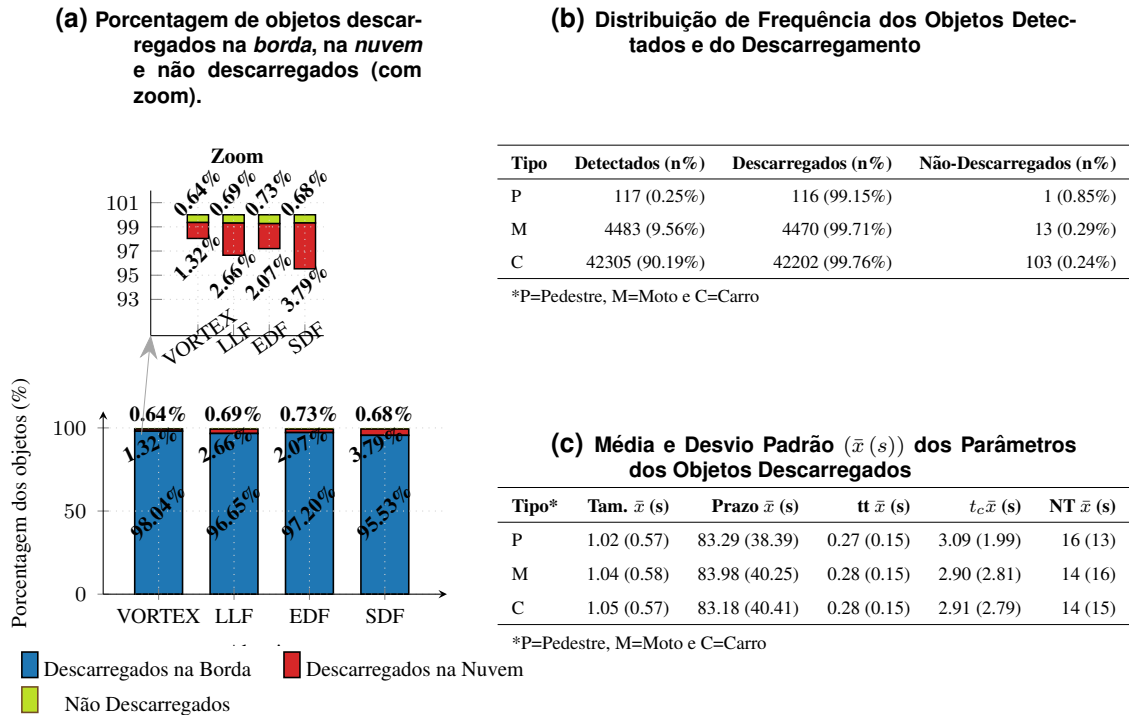
Menor Folga Primeiro (LLF - Least Laxity First). Essas estratégias foram selecionadas por representarem heurísticas fundamentais orientadas a prazo, volume de dados e folga temporal, respectivamente, constituindo referências adequadas para examinar os ganhos introduzidos por uma abordagem sensível ao contexto de rede e mobilidade. Todas as políticas foram executadas sob condições idênticas de carga, topologia e configuração de dispositivos, assegurando comparabilidade e isolando o impacto do algoritmo na dinâmica do sistema.

A análise concentra-se em métricas diretamente associadas à eficiência temporal e operacional do descarregamento. A porcentagem de objetos descarregados quantifica a capacidade efetiva do sistema em completar transmissões dentro das restrições impostas. A folga residual no momento do descarregamento captura o grau de antecipação das decisões, funcionando como um indicador indireto de robustez frente a variações futuras de rede. A latência de descarregamento mede o intervalo entre detecção e conclusão da transferência, refletindo o desempenho fim a fim percebido pela aplicação. Por fim, o número potencial de retransmissões após o descarregamento expressa a margem operacional remanescente, permitindo inferir o quão conservadora ou arriscada foi a decisão de escalonamento.

## 6. Resultados

Com o objetivo de caracterizar o comportamento do descarregamento dos objetos e descrever o perfil dos objetos processados no cenário avaliado, a Figura 3 apresenta: (a) um gráfico de colunas empilhadas com o percentual de objetos estratificado por algoritmo em três categorias: descarregados na borda (azul), descarregados na nuvem (vermelho) e não descarregados (verde). O *zoom* na região superior evidencia que, embora a maior parte dos objetos seja descarregada na borda em todos os algoritmos, observa-se também uma fração encaminhada à nuvem e uma parcela residual que não é descarregada; (b) a distribuição de frequência dos tipos de objetos detectados, descarregados e não descarre-

gados; e (c) a média amostral e o desvio padrão amostral dos principais parâmetros dos objetos descarregados, discriminados por tipo.



**Figura 3. Atendimento do prazo no descarregamento e resumo por tipo de objeto: (a) porcentagem de objetos descarregados na borda, na nuvem e não descarregados (com zoom), (b) frequência de detecção e descarregamento e (c) estatísticas dos parâmetros dos objetos descarregados.**

Em termos comparativos, nota-se, na Figura (a), que o VORTEX apresenta a menor proporção de objetos enviados à nuvem (1,32%) e a menor taxa de não descarregamento (0,64%), enquanto que os demais exibem percentuais progressivamente maiores de descarregamento na nuvem e taxas semelhantes de não descarregamento. Esse comportamento pode ser explicado pela política do VORTEX: ao selecionar tarefas com  $NT > 1$ , o algoritmo tende a priorizar objetos com menor tamanho, aumentando o volume de objetos descarregados, reduzindo a probabilidade de descarregar objetos em condição crítica do prazo e, assim, evitando encaminhamentos para à nuvem.

Na Tabela (b), observa-se que a maior parte dos objetos detectados corresponde a carros (90,19%), enquanto pedestres (0,25%) e motocicletas (9,56%) representam frações menores. Além disso, a parcela de não descarregados é baixa para todos os tipos de objeto, indicando que, independentemente da classe, a maioria dos objetos detectados é efetivamente descarregada.

Na Tabela (c) são apresentadas a média amostral e o desvio padrão amostral ( $\bar{x}(s)$ ) dos parâmetros dos objetos descarregados, estratificados por tipo (Pedestre, Moto e Carro). Observa-se que as estatísticas do tamanho do objeto, do prazo, do tempo de transmissão do objeto ( $tt$ ), do tempo de colaboração  $t_c$  e do número de vezes que o objeto pode ser transmitido, após chegar a borda, ( $NT$ ), permanecem próximas entre as classes, indicando um comportamento relativamente homogêneo entre os tipos de objetos.

O VORTEX apresenta na Figura 4 a maior taxa de atendimento do prazo dos

objetos (99,1%), enquanto o SDF obtém a menor (96,6%). Esse comportamento está associado à política de seleção de cada método.

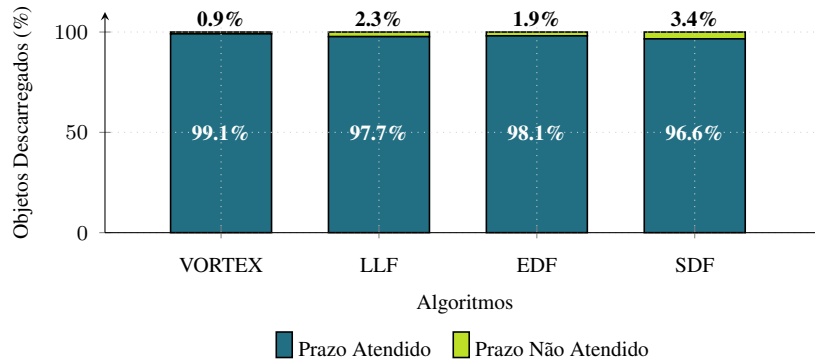


Figura 4. Atendimento do Prazo de Vida dos Objetos Descarregados

Os pontos da Figura 5, representam a folga do *prazo* dos objetos no instante do descarregamento. Observa-se que o VORTEX apresenta a maior proporção de objetos descarregados com *prazo* atendido (99,1%) e, simultaneamente, o menor número de violações (54; 0,9%), o que está alinhado ao objetivo do algoritmo. Em contraste, o SDF concentra a maior fração de objetos com *prazo* não atendido, resultado coerente com sua política de seleção, sem considerar o prazo.

Os algoritmos LLF e EDF apresentam desempenho intermediário e relativamente semelhante: o LLF prioriza objetos com menor folga, enquanto o EDF prioriza aqueles com menor prazo. Entretanto, ambos desconsideram o custo da transmissão (associado ao tamanho do objeto). Assim, um objeto com prazo curto, mas de grande tamanho, pode dominar o atendimento e atrasar o processamento de outros objetos, aumentando a probabilidade de violações do prazo e impactando na latência do descarregamento.

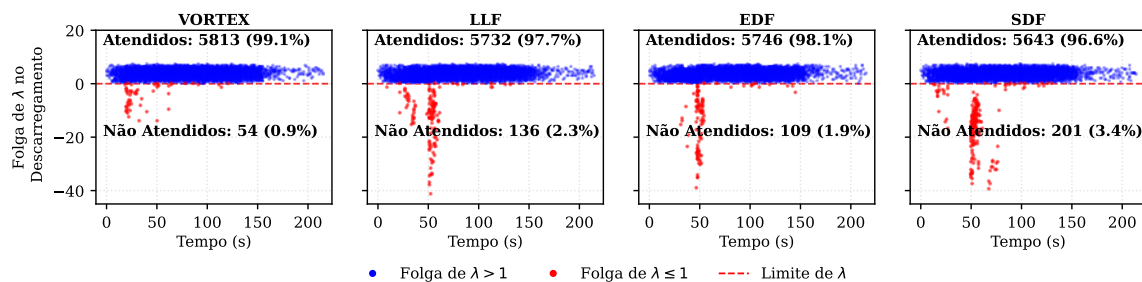
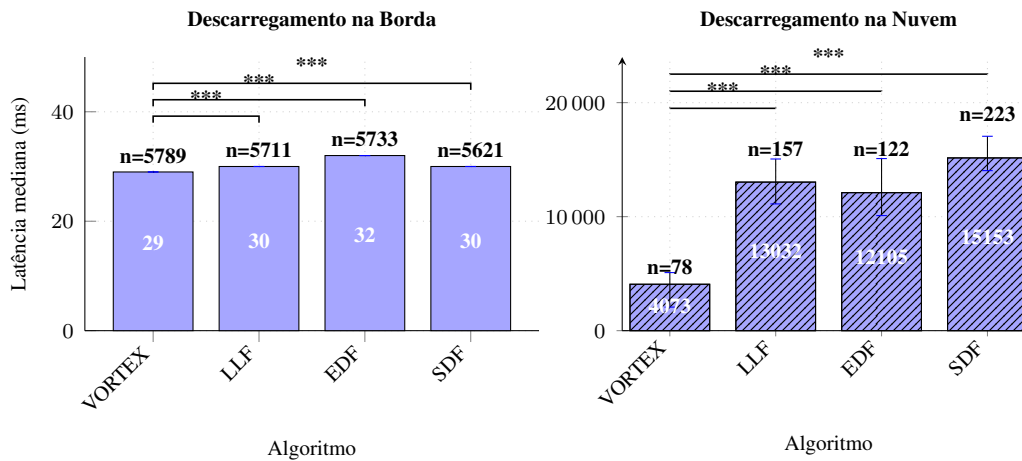


Figura 5. Folga do Prazo no Descarregamento

Para analisar a latência do descarregamento dos objetos, a Figura 6 apresenta a latência mediana (com IC de 95%) e o resultado do teste de Dunn, em dois painéis: à esquerda, dos objetos descarregados na borda; à direita, dos objetos descarregados na nuvem. A mediana foi adotada por se tratar de uma medida robusta à assimetria e à presença de valores extremos, comuns em métricas de latência.

Como esperado, as latências medianas observadas na borda são substancialmente menores do que aquelas observadas na nuvem, refletindo o tempo de espera adicional para o encaminhamento à nuvem. Considerando apenas os objetos descarregados na borda, o VORTEX apresenta a menor latência mediana, com diferenças estatisticamente signifi-



**Figura 6. Latência mediana (ms) [IC 95%] do descarregamento: Borda (esquerda) e Nuvem (direita) com teste de Dunn.**

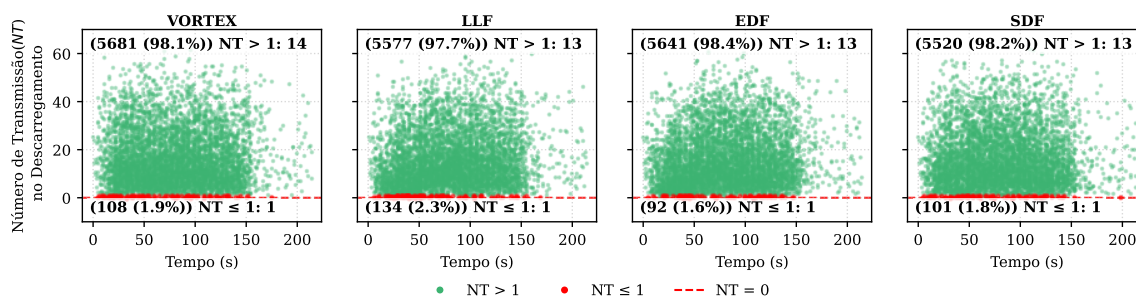
cativas em relação aos demais algoritmos ao nível de 5% (teste de Kruskal–Wallis seguido do pós-teste de Dunn). No gráfico da nuvem, as diferenças tornam-se ainda mais evidentes: o VORTEX mantém latência mediana significativamente inferior aos demais algoritmos ao nível de (5%), o que é consistente com a política de priorização de objetos com maior criticidade temporal e com a consequente redução de espera dos objetos que acabam sendo encaminhadas para a nuvem.

Após o descarregamento na borda, os objetos podem ser retransmitidos para veículos próximos (VECs) a fim de apoiar a percepção cooperativa, especialmente em situações em que um veículo vizinho não conseguiu identificá-los localmente. A Figura 7 apresenta a distribuição do número de transmissões remanescentes do objeto ao chegar à borda ( $NT$ ). Os pontos em vermelho representam objetos com  $NT \leq 1$ , isto é, com margem de retransmissão limitada (no máximo uma transmissão adicional), enquanto os pontos em verde correspondem a  $NT > 1$ , indicando maior folga para retransmissões.

Observa-se que, para todos os algoritmos, a maioria dos objetos chega à borda com  $Ntt > 1$  (aproximadamente 98%), ao passo que uma fração pequena apresenta  $NT \leq 1$  (entre 1,6% e 2,3%, conforme indicado nos painéis). Além disso, considerando os objetos com  $NT > 1$ , a mediana de  $NT$  é semelhante entre os algoritmos: cerca de 14 para o VORTEX e aproximadamente 13 para os demais. Isso sugere que, embora existam objetos com valores elevados de  $NT$  (cauda superior chegando a dezenas de transmissões), o comportamento central (mediano) de  $NT$  na chegada à borda varia pouco entre os algoritmos no cenário avaliado.

## 7. Conclusão

Este artigo apresentou o VORTEX, uma estratégia de descarregamento em nível de objeto projetada para operar ao longo do contínuo borda–nuvem em redes veiculares dinâmicas e sensíveis à latência. Ao substituir a transmissão de fluxos de vídeo completos por uma abordagem orientada a objetos com priorização sensível a prazos, o VORTEX equilibra restrições de latência, capacidade e largura de banda, permitindo decisões de descarregamento mais eficientes e contextualmente informadas. Os resultados demonstraram que a estratégia alcança 99,1% de cumprimento de prazos, reduz encaminhamentos desnecessários aos nós de borda e apresenta melhorias estatisticamente significati-



**Figura 7. Número de Vezes que o Objeto Ainda Pode ser Transmitido ao Chegar à Borda, por Algoritmo**

vas de latência em comparação com políticas clássicas de escalonamento. Esses achados indicam que decisões de descarregamento baseadas em granularidade fina, aliadas à orquestração ao longo do contínuo computacional, constituem um caminho promissor para sustentar aplicações veiculares de tempo crítico e apoiar a percepção cooperativa em ambientes de larga escala.

### Disponibilidade de Artefatos

Em aderência aos princípios da Ciência Aberta, o código-fonte e o dataset utilizados neste trabalho podem ser acessados em: <https://github.com/AnegraMariese10/VORTEX>.

### Agradecimentos

Este estudo foi financiado, em parte, pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001, pela FAPESB, e pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), Brasil, sob a concessão nº 403231/2023-0.

### Referências

- Agarwal, A., Pal, R., and Prakash, A. (2020). A Scheduling Algorithm Including Deadline of Messages in Vehicular Ad Hoc Network. In *Advances in VLSI, Communication, and Signal Processing: Select Proceedings of VCAS 2018*, pages 115–123. Springer.
- Ferreira, J., Freire, M., Cruz, E. M., Prazeres, C. V., Figueiredo, G. B., and Peixoto, M. L. M. (2025). Leading the Way: Reducing Network Traffic in Vehicular Ad Hoc Networks Through Cluster Leader Algorithms. *Ad Hoc Networks*, 178:103932.
- Freire, M., Maia, A., Figueiredo, G., Prazeres, C., Lobato, W., Villas, L., Sommer, C., and Peixoto, M. (2025). Clear Data, Clear Roads: Imputing Missing Data for Enhanced Intersection Flow of Connected Autonomous Vehicles. *Journal of Network and Computer Applications*, 242:104233.
- Kumar, V., Gupta, N., and Mäkelä, J. (2024). Adaptive Priority and Optimum Parameters Based Relay Vehicle Selection Scheme in Vehicular Ad Hoc Networks. *Available at SSRN 4821614*.
- Lee, M., Song, J., Jeong, J., and Kwon, T. (2015). Dove: Data Offloading Through Spatio-Temporal Rendezvous in Vehicular Networks. In *2015 24th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–8. IEEE.

- Maia, A., Freire, M., Mello, T., Rodrigues-Filho, R., Almeida, E., Prazeres, C., Figueiredo, G., and Peixoto, M. (2025). Q-Edge: Leveraging Quantum Computing for Enhanced Software Engineering in Vehicular Networks. In *Proceedings of the 40th ACM/SIGAPP Symposium on Applied Computing*, pages 1457–1467.
- Maresch, M. and Nastic, S. (2024). Vate: Edge-Cloud System for Object Detection in Real-Time Video Streams. In *2024 IEEE 8th International Conference on Fog and Edge Computing (ICFEC)*, pages 27–34. IEEE.
- Meng, J., Tan, H., Xu, C., Cao, W., Liu, L., and Li, B. (2019). Dedas: Online Task Dispatching and Scheduling with Bandwidth Constraint in Edge Computing. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pages 2287–2295. IEEE.
- Nguyen, N. H., Nguyen, V.-D., Nguyen, A. T., Van Thieu, N., Nguyen, H. N., and Chatzinotas, S. (2024). Deadline-Aware Joint Task Scheduling and Offloading in Mobile Edge Computing Systems. *IEEE Internet of Things Journal*.
- Oza, P., Hudson, N., Chantem, T., and Khamfroush, H. (2024). Deadline-Aware Task Offloading for Vehicular Edge Computing Networks Using Traffic Light Data. *ACM Transactions on Embedded Computing Systems*, 23(1):1–25.
- Peixoto, M. L. M. (2024). Quantum Edge Computing for Data Analysis in Connected Autonomous Vehicles. In *2024 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–6. IEEE.
- Qiu, J., Wang, R., Hu, B., Guérin, R., and Lu, C. (2024). Optimizing Edge Offloading Decisions for Object Detection. In *2024 IEEE/ACM Symposium on Edge Computing (SEC)*, pages 164–177. IEEE.
- Raja, G., Ganapathisubramanian, A., Anbalagan, S., Baskaran, S. B. M., Raja, K., and Bashir, A. K. (2020). Intelligent Reward-Based Data Offloading in Next-Generation Vehicular Networks. *IEEE Internet of Things Journal*, 7(5):3747–3758.
- Rajareddy, G. N., Mishra, K., Majhi, S. K., Sahoo, K. S., and Bilal, M. (2025a). M-SOS: Mobility-Aware Secured Offloading and Scheduling in Dew-Enabled Vehicular Fog of Things. *IEEE Transactions on Intelligent Transportation Systems*.
- Rajareddy, G. N., Mishra, K., Majhi, S. K., Sahoo, K. S., and Bilal, M. (2025b). M-SOS: Mobility-Aware Secured Offloading and Scheduling in Dew-Enabled Vehicular Fog of Things. *IEEE Transactions on Intelligent Transportation Systems*.
- Saleem, Y., Mitton, N., and Loscri, V. (2021). Divine: Data Offloading in Vehicular Networks with QoS Provisioning. *Ad Hoc Networks*, 123:102665.
- Silva, C., Magaia, N., and Grilo, A. (2023). Task Offloading Optimization in Mobile Edge Computing Based on Deep Reinforcement Learning. In *Proceedings of the Int'l ACM Conference on Modeling Analysis and Simulation of Wireless and Mobile Systems*, pages 109–118.
- Taherkhani, N. and Pierre, S. (2016). Prioritizing and Scheduling Messages for Congestion Control in Vehicular ad Hoc Networks. *Computer Networks*, 108:15–28.