

Elasticidade de Containers em VANETs: Uma Estratégia de Sustentabilidade Energética em RSUs

Luis Guilherme Silva¹, Israel Cardoso¹, Vandirleya Barbosa¹, Melissa Alves¹,
Leonel Feitosa¹, Paulo A. L. Rego² e Francisco Airton Silva¹

¹Laboratório de Pesquisa Aplicada a Sistemas Distribuídos (PASID),
Universidade Federal do Piauí (UFPI), Picos, PI

²Universidade Federal do Ceará (UFC), Fortaleza, CE

{luis.e, israel.araujo, vandirleya.barbosa}@ufpi.edu.br

{melissaalves, leonelfeitosa, faps}@ufpi.edu.br

paulo@dc.ufc.br

Resumo. Este artigo propõe um modelo de desempenho energético baseado em Redes de Petri Estocásticas (SPNs), que utiliza um sistema de autoscaling dinâmico para ajustar automaticamente a capacidade das RSUs, reduzindo desperdícios de energia sem comprometer a qualidade do serviço. A solução também inclui mecanismos de reinstanciação para lidar com falhas e garantir resiliência operacional. Os resultados mostram configurações eficientes resultaram em uma melhoria de até 25% na eficiência energética. Enquanto a alocação dinâmica adequada possibilitou reduzir o desperdício computacional sem afetar a escalabilidade, podendo tornar redes veiculares mais sustentáveis e eficientes.

Abstract. This paper proposes an energy performance model based on Stochastic Petri Nets (SPNs), which uses a dynamic autoscaling system to automatically adjust the capacity of RSUs, reducing energy waste without compromising quality of service. The solution also includes reinstantiation mechanisms to deal with failures and ensure operational resilience. The results show that efficient configurations resulted in up to a 25% improvement in energy efficiency. While proper dynamic allocation made it possible to reduce computational waste without affecting scalability, potentially making vehicular networks more sustainable and efficient.

1. Introdução

A evolução das Redes Ad Hoc Veiculares (VANETs) trouxe avanços significativos na comunicação entre veículos e infraestruturas, contribuindo para a segurança, eficiência e sustentabilidade do trânsito. Nesse cenário, as *Road Side Units* (RSUs) desempenham um papel central ao processar dados em tempo real e garantir a continuidade dos serviços veiculares [Silva et al. 2023]. No entanto, à medida que sistemas mais complexos são integrados, como monitoramento ambiental, serviços de emergência e aplicações de logística, as RSUs enfrentam a necessidade de equilibrar sua capacidade de processamento com o consumo energético, demandando soluções escaláveis e adaptativas [Elhoseny and Shankar 2020].

Além da crescente complexidade das aplicações, as RSUs devem lidar com variações imprevisíveis na carga computacional causada por eventos extraordinários, como variação no volume de tráfego, acidentes de trânsito, condições climáticas adversas ou aumento súbito no fluxo de informações geradas por veículos conectados [SERPRO 2023]. Nessas situações, a falta de mecanismos de adaptação pode levar à sobrecarga do sistema, degradação do desempenho ou desperdício de recursos durante períodos de baixa atividade. Um dos recursos principais que pode estar sendo desperdiçado é a energia, buscando reduzir o impacto ambiental. Assim, estratégias de *autoscaling* emergem como uma solução indispensável, permitindo que as RSUs ajustem dinamicamente sua capacidade computacional para responder a esses cenários críticos sem comprometer o desempenho e a eficiência energética [Elhoseny et al. 2023].

A implementação de soluções para VANETs em larga escala enfrenta desafios práticos, como o alto custo envolvido na montagem de infraestruturas reais de experimentos [Santos et al. 2024]. A necessidade de testar novas arquiteturas e estratégias de *autoscaling* em ambientes físicos demanda investimentos substanciais em equipamentos, manutenção e monitoramento contínuo [Parashar and Tiwari 2023]. Em contrapartida, a falta de um mecanismo de adaptação eficiente pode resultar em falhas no sistema, como sobrecarga de RSUs, falhas de comunicação e comprometimento da qualidade dos serviços [AlMarshoud et al. 2024]. Nesse contexto, as SPNs se destacam pela capacidade de modelar de forma precisa a dinâmica [Feitosa et al. 2023]. O *autoscaling* em RSUs, considera variáveis como a concorrência de requisições, a adaptação do sistema às mudanças no tráfego e falhas em processos de instancição.

Este trabalho provém de duas pesquisas anteriores, na primeira foi aplicada SPNs para prever o comportamento de um sistema de *autoscaling* em um ambiente de nuvem genérico [Fé et al. 2022]. Na segunda pesquisa [Silva et al. 2024], foi aplicado um sistema de *autoscaling* no contexto de VANETs para mensurar o desempenho do sistema. No presente estudo, a estratégia proposta é adaptada ao contexto das VANETs, com foco na eficiência energética e na sustentabilidade dos sistemas de comunicação veicular. O objetivo é oferecer uma solução adaptativa para o *autoscaling* das RSUs, garantindo não só o desempenho do sistema, mas também a otimização do consumo energético, alinhando a operação das VANETs aos princípios de sustentabilidade. Ao otimizar a alocação de *containers*, o modelo assegura que recursos computacionais sejam utilizados de forma inteligente, reduzindo o desperdício de energia sem comprometer o desempenho do sistema.

O restante deste estudo está estruturado da seguinte maneira: A Seção 2 expõe pesquisas relacionadas por meio de uma tabela comparativa entre os estudos avaliados. A Seção 3 descreve a arquitetura adotada neste estudo. A Seção 4 apresenta o modelo SPN sugerido e as métricas empregadas para a análise de desempenho. Na Seção 5, o resultado do estudo de caso é apresentado. A Seção 6 apresenta as conclusões e discute trabalhos futuras.

2. Trabalhos Relacionados

Esta seção apresenta uma análise dos trabalhos relacionados, considerando estudos que abordam balanceamento de carga e/ou consumo de energia em VANETs. A comparação dos trabalhos foi realizada com base nos seguintes critérios: métricas, foco no consumo

de energia, enfoque no balanceamento de carga, método de avaliação empregado e o foco principal dos estudos.

Os trabalhos analisados utilizam métodos de avaliação como simulação, experimentação e modelagem, cada um com foco específico na validação das soluções propostas. Estudos baseados em simulação, como [Raza et al. 2021] e [Sethi et al. 2020], exploram estratégias como o posicionamento dinâmico de *Unmanned Aerial Vehicle* (UAV) e o escalonamento inter-RSU alimentado por energia renovável, avaliando o desempenho em cenários controlados. Trabalhos experimentais, como [Haider et al. 2024] e [Qun and Arefzadeh 2021], concentram-se na validação prática, incluindo organização de RSUs em clusters e uso de fog computing para balanceamento de carga. Já estudos fundamentados em modelagem, como [Agarwal et al. 2016], propõem estratégias de roteamento que otimizam o consumo energético e prolongam a vida útil da rede. No contexto de redes veiculares, as abordagens incluem arquiteturas assistidas por UAVs, gestão de recursos em RSUs e técnicas de roteamento. Estudos como [Raza et al. 2021] e [Mohammed et al. 2023] destacam o uso de UAVs para melhorar a cobertura e a distribuição de carga em áreas congestionadas. [Rajendran et al. 2023] e [Mehta and Mahato 2023] focam na gestão de RSUs, abordando localização estratégica e redistribuição de tarefas entre servidores de neblina. [Wu et al. 2022] propõe a redistribuição de carga computacional nos veículos, otimizando tempo de resposta, consumo energético e capacidade de armazenamento. Por sua vez, [Agarwal et al. 2016] utiliza estratégias de roteamento para balancear a carga energética e aumentar a conectividade.

As métricas de desempenho analisadas priorizam eficiência energética, escalabilidade e qualidade de serviço. Trabalhos como [Qun and Arefzadeh 2021] e [Agarwal et al. 2016] propõem soluções sustentáveis para otimizar o consumo energético. [Sethi et al. 2020] utiliza métricas como probabilidade de recepção de pacotes e atraso médio para avaliar escalonamento em RSUs. [Wu et al. 2015] foca na redução da latência de *handoff*, enquanto [Haider et al. 2024] e [Mohammed et al. 2023] abordam congestionamento e escalabilidade, destacando a organização de clusters de RSUs e a integração de UAVs para otimização em larga escala. Diferentemente das abordagens tradicionais, que tratam a eficiência energética como uma métrica isolada de consumo de recursos, este estudo adota uma visão mais ampla, integrando a sustentabilidade como um conceito que considera não apenas a redução do consumo elétrico, mas também a pegada de carbono, o impacto ambiental da infraestrutura computacional e a capacidade do sistema de operar de forma eficiente a longo prazo. Enquanto os trabalhos anteriores buscam otimizar a eficiência energética com foco na minimização do consumo imediato, nossa abordagem analisa os efeitos do autoscaling dinâmico na sustentabilidade operacional da rede, garantindo que a adaptação dos recursos computacionais contribua para um uso mais racional da infraestrutura. Ao abordar o balanceamento de carga, o desempenho e a eficiência energética em VANETs sob essa ótica sustentável, o modelo proposto diferencia-se por empregar SPNs como ferramenta central para a modelagem e análise de cenários dinâmicos. Isso permite validar a eficácia da solução sem os altos custos experimentais de testes em larga escala. Os resultados obtidos demonstraram ganhos em eficiência e escalabilidade, ao mesmo tempo que reforçam a viabilidade da abordagem para reduzir desperdícios computacionais e promover maior sustentabilidade em ambientes urbanos inteligentes. Dessa forma, além de preencher algumas lacunas existentes na

literatura, este trabalho oferece novas perspectivas para a criação de padrões que otimizem redes veiculares de maneira eficiente e ambientalmente responsável.

3. Arquitetura

Esta Seção apresenta a arquitetura proposta, utilizada como base para a criação do modelo SPN, adota um sistema de autoscaling dinâmico baseado em containers. O sistema busca otimizar o consumo de energia das RSUs, garantindo um equilíbrio entre eficiência energética e qualidade do serviço veicular. O sistema ajusta a alocação de recursos computacionais de acordo com a demanda. Isso evita desperdício de energia em períodos de baixa carga e assegura capacidade suficiente em momentos de pico.

Esse ajuste adaptativo permite que as RSUs operem sustentavelmente, reduzindo o impacto ambiental sem comprometer a continuidade do processamento das requisições. A análise detalhada do impacto energético do autoscaling, permitindo prever cenários de otimização e garantir que a infraestrutura veicular seja dimensionada de maneira eficiente. Dessa forma, a arquitetura não apenas promove um uso racional da energia, mas também contribui para a sustentabilidade das redes veiculares, alinhando-se aos requisitos de eficiência e resiliência demandados por ambientes urbanos inteligentes.

O sistema é composto por três módulos interdependentes, cada um desempenhando um papel fundamental na gestão eficiente dos recursos computacionais das RSUs. O primeiro módulo é responsável pelo monitoramento contínuo, coletando métricas de uso e analisando padrões de tráfego para identificar a necessidade de ajuste na alocação de containers. O segundo módulo implementa o autoscaling adaptativo, garantindo que a infraestrutura computacional se ajuste dinamicamente à demanda, evitando tanto o consumo excessivo quanto a subutilização de recursos. Já o terceiro módulo foca na resiliência do sistema, assegurando que falhas na instânciação ou sobrecarga sejam rapidamente mitigadas por meio da realocação de containers, preservando a estabilidade da rede e a qualidade do serviço veicular.

4. Modelo SPN

Esta seção detalha o modelo desenvolvido, estruturado em quatro módulos principais, que representam as operações do sistema: (1) Admissão, (2) Unidade RSU, (3) Sistema de *autoscaling*, e (4) Sistema de Reinstanciação. Esses módulos foram projetados para analisar o desempenho do sistema em relação à dinamicidade de tráfego e demanda computacional, respeitando os parâmetros da arquitetura adotada.

4.1. Estrutura do Modelo

A Figura 1 ilustra a estrutura principal do modelo. O módulo de Admissão controla o fluxo inicial das requisições. Nele, o lugar T_{ad} representa a entrada de requisições geradas pelo tráfego, enquanto o lugar P_{pqe} organiza a transferência das requisições para a próxima etapa. Em seguida, o módulo RSU processa as solicitações pendentes. As requisições em fila são armazenadas no lugar P_{pqe} até que recursos suficientes estejam disponíveis no lugar P_{caprsu} , indicando *containers* prontos para execução. A transição T_{rsu} transfere a solicitação para o lugar P_{prsu} , onde ocorre o processamento. Após o processamento, a transição T_{trsu}$ devolve os *containers* ao lugar P_{caprsu} , permitindo seu uso em novas operações.

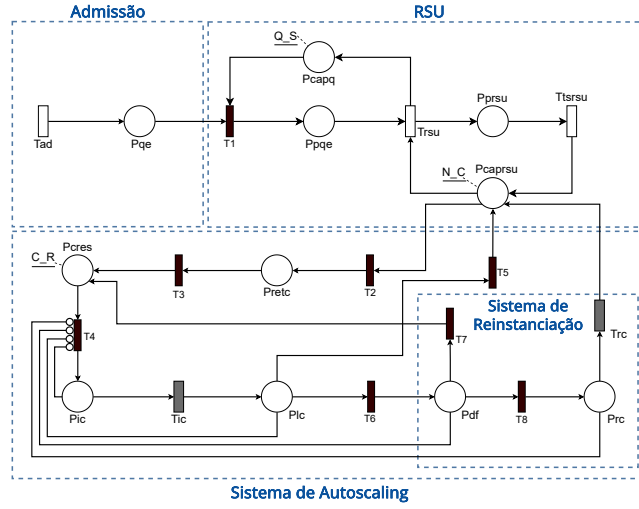


Figura 1. Modelo SPN desenvolvido a partir da arquitetura adotada.

O módulo de *autoscaling* gerencia dinamicamente a capacidade do sistema com base na demanda. Quando é identificada uma alta fila de requisições e redução de capacidade para atender novas requisições, a transição T4 é ativada, acionando a instanciação de novos *containers* a partir do lugar P_{cres}. Esses *containers* passam pelo lugar P_{ic}, onde o tempo necessário para a instanciação é simulado pela transição T_{ic}, antes de serem enviados para o lugar P_{caprsu}, através da transição T5. Por outro lado, quando a demanda diminui, *containers* ociosos são liberados pela transição T2, armazenados temporariamente em P_{retc} e realocados ao reservatório inicial P_{cres} por meio da transição T3.

Por fim, o módulo de Reinstalação lida com falhas durante o processo de instanciação, garantindo a operação contínua do sistema. Quando ocorre uma falha, a transição T6 redireciona o *container* ao lugar de detecção P_{df}. Caso a falha não seja resolvida, o *container* retorna ao reservatório inicial P_{cres} via transição T7. Se a falha for corrigida, a transição T8 encaminha o *container* ao lugar P_{rc}, onde a transição determinística T_{rc} finaliza o processo, realocando o *container* ao lugar de capacidade P_{caprsu}. Essa abordagem fortalece a resiliência e a eficiência operacional do sistema.

4.2. Métricas do Modelo

No contexto deste trabalho, as métricas estão focadas em avaliar o impacto energético do sistema. As métricas utilizadas nos experimentos foram a Energia Consumida (EC), Pegada de Carbono (PC) e Eficiência Energética (EE). Cada uma dessas métricas é essencial para mensurar o desempenho em termos de consumo de recursos e impacto ambiental, considerando o comportamento do sistema sob diferentes condições de carga e operação.

A Equação 1 contém a métrica de EC, que visa calcular o total de energia utilizado pelos componentes ativos do sistema durante a operação. Para isso, o cálculo leva em conta a energia consumida tanto pela RSU quanto pelo *container* de virtualização (Cont), considerando o tempo total de operação. O cálculo é feito tendo como base a utilização do sistema, verificando a esperança de haver tokens em P_{pqsu} e P_{caprsu}, ou seja, requisições sendo processadas pela RSU. E_{OP_RSU} é a energia necessária para

manter a RSU ativa, e E_{OP_Cont} é a energia necessária para manter o *container* em funcionamento. O fator $Time$ representa a duração do tempo em que esses componentes estão ativos e processando informações. O cálculo considera a energia total consumida por hora pelos componentes enquanto eles estão operando simultaneamente.

$$EC = ((E\{\#P_{prsu}\} \times E_{OP_RSU} + E\{\#P_{caprsu}\} \times E_{OP_CONT}) \times Time) \quad (1)$$

A métrica de Pegada de Carbono (PC), definida na Equação 2, quantifica o impacto ambiental do consumo de energia, considerando o fator de emissão de carbono (FEC). Esse fator varia conforme a origem da energia utilizada, refletindo a intensidade de emissões de CO₂ associadas à geração elétrica. No contexto deste estudo, a PC vai além da simples conversão de energia consumida em emissões, pois avalia como a gestão dinâmica dos recursos computacionais influencia a sustentabilidade ambiental. Isso se dá, pois a maioria dos meios de geração de energia gera algum resíduo ou impacto no meio ambiente. O modelo proposto busca não apenas reduzir o consumo energético, mas também minimizar a necessidade de uso de infraestruturas computacionais subutilizadas, o que, por consequência, reduz a pegada de carbono. Assim, essa métrica permite uma avaliação mais abrangente do impacto ambiental da solução, diferenciando eficiência energética.

$$PC = FEC \times EC \quad (2)$$

A Equação 3 define a métrica de Eficiência Energética (EE), que avalia a relação entre a quantidade de tarefas processadas com sucesso e a energia consumida pelo sistema. Essa métrica busca quantificar o uso eficiente dos recursos computacionais, refletindo a capacidade do sistema de realizar operações com o menor consumo energético. Para isso, considera-se a taxa média de requisições em processamento, representada pelo número de tokens no lugar de processamento da RSU P_{prsu} , dividida pelo tempo médio de serviço das requisições T_{tsrsu} . O valor resultante é então normalizado pelo consumo total de energia, permitindo mensurar a eficiência energética da infraestrutura em diferentes cenários operacionais. Essa métrica é fundamental para avaliar a sustentabilidade do sistema e sua capacidade de adaptação a variações na demanda computacional.

$$EE = \left(\left(\frac{P_{prsu}}{T_{tsrsu}} \right) \div EC \right) \times 100 \quad (3)$$

5. Estudo de Caso

Para a implementação e execução dos experimentos, utilizou-se a ferramenta Mercury [Maciel et al. 2017], na versão 5.0.1. Dado que o modelo envolve múltiplas variáveis estocásticas e estados interdependentes, sua complexidade exige simulações com um número elevado de eventos para garantir resultados estatisticamente representativos. Assim, os experimentos foram conduzidos utilizando simulações estacionárias, permitindo a observação do comportamento do sistema em regime permanente, minimizando os efeitos das condições iniciais.

A margem de erro aproximada de 2% foi estabelecida com base no critério de convergência das métricas analisadas, garantindo precisão nos resultados sem comprometer a viabilidade computacional das simulações. Altas margens de erro poderiam comprometer a confiabilidade das estimativas, enquanto margens menores exigiriam um aumento significativo no tempo de execução e confiabilidade das simulações. Os valores dos componentes do sistema foram baseados em fontes acadêmicas anteriormente publicadas, com maior destaque, os estudos [Silva et al. 2024, Fé et al. 2022].

Nesta seção, é realizada uma análise dos resultados de desempenho energético decorrentes da variação da quantidade de *containers* no sistema. Ao longo da análise feita, foi observado que a quantidade de *containers* aparece como um dos principais fatores. Com isso, foi utilizado o sistema de *autoscaling* para variar a quantidade de *containers* disponíveis. A Figura 2 apresenta os resultados dessa variação da quantidade de *containers* em busca das melhores configurações, assim, promovendo uma maior sustentabilidade e o desempenho energético do sistema.

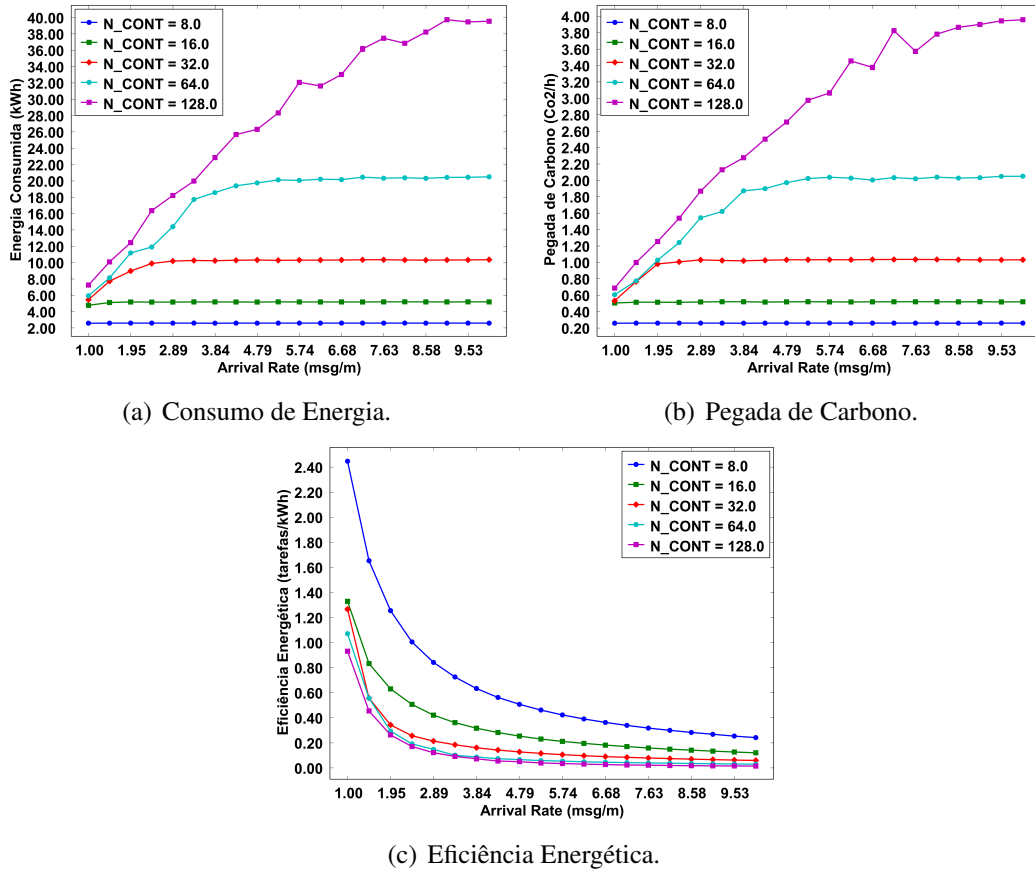


Figura 2. Impactos Energéticos do *autoscaling* na variação da quantidade de *containers*.

A Figura 2(a) apresenta a relação entre o consumo de energia (em kWh) e a taxa de chegada de mensagens (msg/m) para diferentes valores de N_CONT, representando o número de *containers* no sistema. Os resultados mostram que o consumo de energia aumenta significativamente com o número de *containers*, especialmente em taxas de chegada mais altas. Para N_CONT=8, o consumo é o menor entre os cenários, mantendo-se

constante e inferior a 0,4kWh, mesmo com taxas de chegada elevadas. O cenário com $N_CONT=16$ apresenta consumo um pouco maior, mas estável em torno de 0,6kWh, enquanto para $N_CONT=32$, o consumo sobe ligeiramente, estabilizando-se em aproximadamente 0,8kWh. Nos casos de $N_CONT=64$ e $N_CONT=128$, observa-se um comportamento distinto. Com $N_CONT=64$, o consumo inicial é de cerca de 1,0kWh, crescendo gradualmente até se estabilizar em torno de 1,8kWh. Já o cenário com $N_CONT=128$ registra o maior consumo, começando em 1,5kWh e aumentando quase linearmente até 3,5kWh nas taxas mais altas. Os resultados indicam que o aumento no número de *containers* está diretamente associado a um maior consumo de energia. Em cenários com baixa demanda (taxas reduzidas de msg/m), a utilização de um menor número de *containers* pode ser mais eficiente em termos de consumo energético. Contudo, em situações de alta demanda, o aumento do consumo é inevitável devido à necessidade de maior capacidade para lidar com a carga de requisições.

A Figura 2(b) examina a pegada de carbono (em CO₂/h) em relação à taxa de chegada de mensagens (msg/m) para diferentes quantidades de *containers*. Para $N_CONT=8$, a pegada de carbono mantém-se estável em torno de 0,2 CO₂/h, indicando eficiência ambiental com menos *containers*. Para $N_CONT=16$ e $N_CONT=32$, há um leve aumento conforme a taxa de chegada cresce, variando entre 0,4 e 0,8 CO₂/h. Com $N_CONT=64$, o impacto ambiental cresce de forma mais significativa, superando 2 CO₂/h quando a taxa de chegada ultrapassa 5 msg/m. No caso de $N_CONT=128$, observa-se um aumento acentuado, chegando a 4 CO₂/h nas maiores taxas, evidenciando um crescimento exponencial nas emissões. Embora configurações com muitos *containers*, como $N_CONT=128$, sejam adequadas para altas taxas de chegada, isso ocorre à custa de um impacto ambiental consideravelmente maior. Já configurações com menos *containers*, como $N_CONT=8$, são mais sustentáveis, mas podem não atender à demanda em cenários de alta carga. Dessa forma, é necessário equilibrar desempenho e sustentabilidade ambiental.

A Figura 2(c) apresenta os resultados de eficiência energética em tarefas/kWh para diferentes configurações de *containers*. Para $N_CONT=8$, a eficiência é maior em baixas taxas de chegada, iniciando em 2,4 tarefas/kWh e reduzindo para menos de 0,4 tarefas/kWh em taxas superiores a 5 msg/m. Com $N_CONT=16$, a eficiência inicial é menor, começando em 1,2 tarefas/kWh e estabilizando-se próximo de 0,4 tarefas/kWh em altas taxas. Nos casos de $N_CONT=32$, 64 e 128, os valores iniciais de eficiência são ainda mais baixos, caindo rapidamente para menos de 0,2 tarefas/kWh em taxas intermediárias. Os resultados indicam que a eficiência energética diminui à medida que o número de *containers* aumenta, especialmente em cenários de alta carga. Embora configurações com poucos *containers*, como $N_CONT=8$, sejam mais eficientes energeticamente, elas podem não atender a demandas elevadas. Por outro lado, configurações com muitos *containers*, como $N_CONT=128$, têm baixa eficiência energética, mesmo em condições de alta carga, refletindo um uso menos sustentável de energia.

6. Conclusão

Este artigo apresentou um modelo baseado em SPN para analisar o impacto do *autoscaling* dinâmico em VANETs, otimizando o consumo de energia das RSUs sem comprometer a qualidade do serviço. Os experimentos demonstraram que a adaptação dinâmica dos *containers* reduz significativamente o consumo energético e a pegada de carbono, garantindo o equilíbrio entre eficiência energética e escalabilidade. Configurações oti-

mizadas podem aumentar a eficiência em até 25%, enquanto configurações inadequadas resultam em desperdício de energia e maior pegada de carbono. Além disso, os resultados evidenciaram um *trade-off* entre sustentabilidade e capacidade de atendimento, destacando que configurações menores de *containers* são mais eficientes energeticamente, mas podem ser insuficientes para cargas elevadas. A modelagem permitiu observar que a alocação dinâmica reduz desperdícios computacionais e melhora a resiliência operacional das RSUs, assegurando a continuidade dos serviços em cenários de variação abrupta da demanda. Embora a abordagem proposta tenha se mostrado eficaz, desafios como o custo computacional do monitoramento contínuo e possíveis gargalos de comunicação ainda precisam ser mitigados. Trabalhos futuros explorarão aprendizado de máquina para prever variações na carga e arquiteturas distribuídas para otimizar a orquestração das RSUs, além da integração de estratégias preditivas de escalonamento, buscando maior eficiência energética e adaptação do sistema a cenários urbanos dinâmicos.

Referências

- Agarwal, S., Das, A., and Das, N. (2016). An efficient approach for load balancing in vehicular ad-hoc networks. In *2016 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pages 1–6.
- AlMarshoud, M., Sabir Kiraz, M., and H. Al-Bayatti, A. (2024). Security, privacy, and decentralized trust management in vanets: a review of current research and future directions. *ACM Computing Surveys*, 56(10):1–39.
- Elhoseny, M., El-Hasnony, I. M., and Tarek, Z. (2023). Intelligent energy aware optimization protocol for vehicular adhoc networks. *Scientific Reports*, 13(1):9019.
- Elhoseny, M. and Shankar, K. (2020). Energy efficient optimal routing for communication in vanets via clustering model. *Emerging technologies for connected internet of vehicles and intelligent transportation system networks: Emerging technologies for connected and smart vehicles*, pages 1–14.
- Fé, I., Matos, R., Dantas, J., Melo, C., Nguyen, T. A., Min, D., Choi, E., Silva, F. A., and Maciel, P. R. M. (2022). Performance-cost trade-off in auto-scaling mechanisms for cloud computing. *Sensors*, 22(3):1221.
- Feitosa, L., Rego, P. A., and Silva, F. A. (2023). Avaliação de desempenho de migração ao vivo de contêineres com redes de petri estocásticas. In *Anais do XXIV Workshop de Testes e Tolerância a Falhas*, pages 94–107. SBC.
- Haider, S. E., Khan, M. F., and Saeed, Y. (2024). Adaptive load balancing approach to mitigate network congestion in vanets. *Computers*, 13(8).
- Maciel, P., Matos, R., Silva, B., Figueiredo, J., Oliveira, D., Fé, I., Maciel, R., and Dantas, J. (2017). Mercury: Performance and dependability evaluation of systems with exponential, expolynomial, and general distributions. In *2017 IEEE 22nd Pacific Rim international symposium on dependable computing (PRDC)*, pages 50–57. IEEE.
- Mehta, T. and Mahato, D. P. (2023). Effective scheduling and nature inspired hybrid load balancing in vanets. In *8th International Conference on Computing in Engineering and Technology (ICCET 2023)*, volume 2023, pages 266–273.

- Mohammed, N. A., Almutoki, S. M. M., Mansoor, R., Jaber, A. K., Ghzawi, B. A. H. K. A., and Alsalamy, A. H. (2023). Resource allocation with energy balancing for uavs assisted vanets based intelligent transportation system. In *2023 Al-Sadiq International Conference on Communication and Information Technology (AICCIT)*, pages 282–287.
- Parashar, S. and Tiwari, R. (2023). Traffic control and qos improvement analysis in v-to-v and v-to-rsu communication in vanet. In *2023 World Conference on Communication & Computing (WCONF)*, pages 1–5. IEEE.
- Qun, R. and Arefzadeh, S. M. (2021). A new energy-aware method for load balance managing in the fog-based vehicular ad hoc networks (vanet) using a hybrid optimization algorithm. *IET Communications*, 15(13):1665–1676.
- Rajendran, N., Kumar, R. P., Kavitha, T., Jayakumar, T., and Nanammal, V. (2023). Delay-tolerant load balancing routing model for rsu-assisted distributed vehicular adhoc networks. In *2023 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES)*, pages 1–7.
- Raza, A., Bukhari, S. H. R., Aadil, F., and Iqbal, Z. (2021). An uav-assisted vanet architecture for intelligent transportation system in smart cities. *International Journal of Distributed Sensor Networks*, 17(7):15501477211031750.
- Santos, L., Brito, C., Fé, I., Carvalho, J., Torquato, M., Choi, E., Min, D., Lee, J.-W., Nguyen, T. A., and Silva, F. A. (2024). Event-based moving target defense in cloud computing with vm migration: A performance modeling approach. *IEEE Access*.
- SERPRO (2023). Registro nacional de acidentes e estatísticas de trânsito. <https://www.gov.br/transportes/pt-br/assuntos/transito/arquivos-senatran/docs/reanaest>. Acesso em: 28 nov. 2024.
- Sethi, V., Pal, S., and Vyas, A. (2020). Online energy-efficient scheduling algorithm for renewable energy-powered roadside units in vanets. In *2020 IEEE 17th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, pages 506–514.
- Silva, L., Brito, C., Cardoso, I., Sabino, A., Lima, L., Gonçalves, G., Filho, G. R., Fé, I., and Silva, F. (2024). Desvendando a elasticidade de máquinas virtuais em vanets: Uma estratégia para aperfeiçoar o planejamento de capacidade em rsus. In *Anais do XLII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 169–182, Porto Alegre, RS, Brasil. SBC.
- Silva, L. G., Cardoso, I., Brito, C., Barbosa, V., Nogueira, B., Choi, E., Nguyen, T. A., Min, D., Lee, J. W., and Silva, F. A. (2023). Urban advanced mobility dependability: A model-based quantification on vehicular ad hoc networks with virtual machine migration. *Sensors*, 23(23):9485.
- Wu, T.-Y., Obaidat, M. S., and Chan, H.-L. (2015). Qualityscan scheme for load balancing efficiency in vehicular ad hoc networks (vanets). *Journal of Systems and Software*, 104:60–68.
- Wu, Y., Wu, J., Chen, L., Yan, J., and Han, Y. (2022). Load balance guaranteed vehicle-to-vehicle computation offloading for min-max fairness in vanets. *IEEE Transactions on Intelligent Transportation Systems*, 23(8):11994–12013.