Desvendando a Elasticidade de Máquinas Virtuais em VANETs: Uma Estratégia para Aperfeiçoar o Planejamento de Capacidade em RSUs

Luis Guilherme Silva[1], Carlos Brito[1], Israel Cardoso[1], Arthur Sabino [1], Luiz Nelson Lima[1], Glauber Gonçalves[1], Geraldo P. Rocha Filho[2], Iure Fé[1] e Francisco Airton Silva[1]

[1]Universidade Federal de Piauí – PI – Brasil

[2]Universidade Estadual do Sudoeste da Bahia– BA – Brasil

{luis.e,carlosvictor,israel.araujo,arthursabino,

luizznelson,ggoncalves,iure.fe,faps}@ufpi.edu.br

geraldo.rocha@uesb.edu.br

Resumo. *Este artigo apresenta um modelo de desempenho projetado para avaliar a eficácia de um sistema de auto-escalonamento de máquinas virtuais aplicado ao monitoramento em rodovias sujeitas a elevada variabilidade de tráfego com VANETs. Para isso, foi utilizada uma abordagem baseada em redes de Petri estocásticas, capaz de capturar uma variedade de comportamentos distintos associados ao sistema de auto-escalonamento. Os resultados obtidos revelam a importância da quantidade já alocada de máquinas virtuais no sistema inicialmente. Ainda, constatou-se que a aplicação efetiva da estratégia de auto-escalonamento e reinstanciação desempenhou um papel significativo na otimização do desempenho global do sistema.*

Abstract. *This paper presents a performance model designed to evaluate the effectiveness of a virtual machine auto-escalonamento system applied to monitoring on highways subject to high traffic variability with VANETs. For this, an approach based on stochastic Petri nets was used, capable of capturing a variety of distinct behaviors associated with the auto-escalonamento system. The results obtained reveal the importance of the quantity of virtual machines already allocated in the system initially. Furthermore, it was found that the effective application of the auto-escalonamento and reinstantiation strategy played a significant role in optimizing the overall performance of the system.*

1. Introdução

Existe uma necessidade crescente de desenvolver sistemas que aprimorem a eficiência e segurança do tráfego de veículos, oferecendo aos usuários uma variedade de informações sobre as condições viárias e serviços relacionados [Naresh et al. 2024]. As Redes Ad Hoc Veiculares (VANET) são redes sem fio voltadas a proporcionar comunicação entre veículos, oferecendo maior conforto e segurança a todos os envolvidos no trânsito. As VANETs podem ser aplicadas em diversas áreas relacionadas ao controle de tráfego, como segurança de redes [Karabulut et al. 2023], controle de tráfego [Parashar and Tiwari 2023], gerenciamento de vagas de estacionamento

[Ouhmidou et al. 2023], entre outras. Nas redes VANETs, os dados são normalmente processados em estações próximas chamadas de Road Side Units (RSUs).

As RSUs atuais possuem alta capacidade de processamento, porém, a demanda por processamento não é igual ao longo do dia. No período da noite, por exemplo, o tráfego tende a cair e assim não necessitando total disponibilidade da RSU. Para sanar esta questão, existe a opção de usar virtualização através da instanciação e desinstanciação de Máquinas Virtuais (VMs) nestas RSUs, até mesmo focando em migrá-las, como abordamos em um trabalho anterior [Silva et al. 2023]. Tal estratégia adaptativa pode ocorrer de forma automática via auto-escalonamento conforme a demanda. A inteligência por trás disso pode ser simplesmente definir um nível de utilização máximo e mínimo que dispararia o auto-escalonamento.

Ao se implantar novas arquiteturas VANETs, precisa-se saber com antecedência qual a capacidade das RSUs atenderia a demanda mesmo utilizando o auto-escalonamento. Experimentos prévios em ambientes reais demandam recursos financeiros e operacionais significativos. Por isso, modelos analíticos são atrativas alternativas, a exemplo das redes de Petri estocásticas (SPNs) [Bobbio et al. 1998]. Os modelos SPNs destacam-se por sua capacidade representativa, sendo considerados mais intuitivos em comparação com abordagens convencionais. SPNs permitem modelar de forma mais eficaz conceitos complexos, tais como concorrência, paralelismo e sincronização em sistemas dinâmicos. Em um trabalho anterior, por exemplo, exploramos com SPN a questão da disponibilidade de recursos em RSUs [Araújo et al. 2021]. Em outro [Rodrigues et al. 2021], analisamos o desempenho de processamento de RSUs cooperativas em cluster.

Em outro trabalho anterior [Fé et al. 2022a], também usando SPNs, mostramos ser possível prever o comportamento de um mecanismo de auto escalonamento em um ambiente de nuvem genérico. Estendendo tal estratégia, também usando SPN, agora neste artigo exploramos duas novas funcionalidades principais: alteração de taxa de chegada de acordo com período do dia e inclusão da possibilidade de haver falhas ao fazer novas instanciações de máquinas virtuais com respectivas novas tentativas. Assim, podemos resumir as principais contribuições desta pesquisa da seguinte forma:

- **Elaboração de um mecanismo de auto escalonamento em um sistema de monitoramento de rodovias.** O modelo incorpora auto escalonamento, adaptando a capacidade de processamento conforme as variações na demanda de tráfego em diferentes momentos do dia.
- **Avaliar a capacidade do sistema em lidar com falhas de instanciação.** O modelo integra um mecanismo de reinstanciação para recuperar VMs falhadas na instanciação, assegurando adaptabilidade e continuidade operacional.
- **Adaptar diferentes parâmetros em distintos estudos de caso.** O modelo possibilita ajustar parâmetros, como a capacidade de enfileiramento de requisições e a variação na capacidade inicial da RSU, permitindo a realização de diversos estudos de caso pré-implementação do sistema.

A estrutura deste trabalho está organizada da seguinte forma. A Seção 2 apresenta os trabalhos relacionados a este estudo. A Seção 3 descreve a arquitetura base que deu origem ao modelo proposto. A Seção 4 detalha como foi desenvolvido o modelo proposto.

A Seção 5 apresenta estudos de caso de utilização do modelo. Finalmente, a Seção 6 apresenta as conclusões e os trabalhos futuros.

2. Trabalhos Relacionados

Esta seção apresenta os trabalhos relacionados que tratam de gerenciamento de tráfego utilizando alocação de recursos com VMs. Sete artigos relacionados foram elencados e descritos a seguir. [Tang et al. 2019] propõem um esquema de roteamento centralizado com previsão de mobilidade para VANETs, utilizando inteligência artificial e *Software Defined Networks* (SDN). O controlador SDN realiza previsões precisas, permitindo estimativas de probabilidade de transmissão bem-sucedida e atraso médio. Com base nisso, as RSUs calculam rotas ideais, minimizando atrasos. [Silva et al. 2023] propõem o uso de SPN para avaliar arquiteturas de Comunicação e Controle de Veículos (VCC) em VANETs para mobilidade urbana avançada (UAM). O estudo de caso explora disponibilidade e confiabilidade, orientando otimizações para UAM. [Cumbal et al. 2019] propõem uma implantação eficiente de recursos em unidades à beira da estrada (RSUs) usando comunicações heterogêneas para cobrir redes veiculares dinâmicas. Cenários realistas são configurados com ferramentas de mobilidade, e a pesquisa determina a localização ideal da infraestrutura de comunicações para otimizar a cobertura. A atribuição dinâmica de recursos é adaptada a restrição de capacidade e cobertura, buscando o uso ideal da rede veicular.

[Wu et al. 2020] propõem otimizar atribuição de tarefas e alocação de recursos considerando a rápida mudança na topologia das redes veiculares. Formula o problema como uma otimização Min-Max para reduzir a latência total das tarefas, decompondo em sub-problemas. [Siddiqi et al. 2020] abordam integrar veículos com computação em nuvem para lidar com o processamento em tempo real de dados multimídia gerados por veículos inteligentes. Os autores propõem um framework eficiente de alocação de recursos e computação para superar desafios como custo de recursos, tempo de resposta rápido e qualidade da experiência. [Macêdo et al. 2022] propõem um modelo de Stochastic Petri Net para VANETs com Road Side Units (RSU). Neste trabalho foram avaliadas métricas como tempo de resposta e probabilidade de descarte, destacando a subutilização da camada Edge RSU em condições específicas. O estudo destaca a importância da simulação para avaliar configurações e otimizações em VANETs, dadas suas características críticas e custos elevados. [Martin-Faus et al. 2018] abordam o desafio de desenvolver modelos analíticos para analisar o comportamento de VANETs. Destaca a importância de métodos adaptativos para diversos algoritmos e serviços em VANETs, visando manter um desempenho desejado. O foco está na medida de tempo ocioso (Tidle) entre dois nós VANET, abordando sua caracterização por meio de um modelo analítico baseado em uma cadeia de recompensas de Markov.

A Tabela 1 compara pontos importantes deste estudo com outros estudos na área. O primeiro critério comparativo é **Método de avaliação**. Os estudos pesquisados mostraram uma leve tendência para a utilização de modelos SPN para avaliar as propostas. A modelagem SPN tende a ser bem aproveitada em situações onde o sistema proposto exigiria um alto custo para testes reais. Sendo assim, nosso estudo utiliza modelagem SPN para conseguir de maneira probabilística predizer algumas situações de desempenho que possam acontecer e propiciar uma adaptabilidade em escolhas de componentes para um sistema real. As **Métricas** avaliadas nesse contexto podem ser diversas, contudo as

métricas de desempenho se tornam cruciais para um sistema desse tipo. O presente estudo foca em métricas de desempenho como: (número de VMs em Uso) NVMU, (drop probability) DP, (troughput) TP, (tempo médio de resposta) MRT e utilização (U). As métricas estudadas ajudam a entender os comportamentos do sistema e parametrizar o mecanismo de auto-escalonamento. Diferente de todos os outros estudos, nosso modelo apresenta um **Tipo de alocação de recursos** de auto-escalonamento. Os demais estudos apresentam tipo de alocação padrão (manualmente), diferindo da nossa proposta que é escalonar dinamicamente os recursos conforme demanda. Outro fator comparado é se o estudo **Considera variações de tráfego**. Apenas 2 trabalhos não consideram fluxo de tráfego. Detalhe importante, ao considerar as variações de fluxo de tráfego ao longo do dia, nosso trabalho considera adaptar a capacidade de processamento conforme as variações e demandas de tráfego. O ultimo fator comparado é se o estudo **Considera reinstanciação de VMs**. No levantamento realizado, não foi encontrado outro trabalho que buscasse a implementação de um mecanismo de reinstanciação em caso de falha durante o instanciação.

Tabela 1. Trabalhos relacionados.

Trabalho	Método de avaliação	Métricas	Uso de Auto-Escalonamento	Considera variações de tráfego	Considera reinstanciação de VMs
[Tang et al. 2019]	Modelo matemático	Probabilidade de transmissão e atraso médio	Não	Sim	Não
[Silva et al. 2023]	Modelo SPN	Confiabilidade e disponibilidade	Não	Não	Não
[Cumbal et al. 2019]	Modelo ILP	Taxa de cobertura e taxa de comunicação	Não	Sim	Não
[Wu et al. 2020]	Modelo VFC	Atraso total de tempo	Não	Sim	Não
[Siddiqi et al. 2020]	Modelo de filas	Tempo de resposta, custo de recursos e qualidade de serviço	Não	Sim	Não
[Macêdo et al. 2022]	Modelo SPN	MRT, DP e utilização	Não	Não	Não
[Martin-Faus et al. 2018]	Modelo Markov Chain	Tempo ocioso do canal	Não	Sim	Não
Este trabalho	Modelo SPN	NVMU, DP, TP, MRT e U	Sim	Sim	Sim

3. Arquitetura Base

Esta seção apresenta a arquitetura adotada como base para a criação do modelo SPN em questão. A Figura 1 apresenta a arquitetura do sistema, que serviu como base para o modelo. Assim, tem como objetivo principal monitorar de forma eficiente uma rodovia, adaptando-se dinamicamente às variações de tráfego ao longo do dia. O diferencial do sistema reside no mecanismo de auto escalonamento implementado na RSU, que ajusta a capacidade de processamento de acordo com a demanda. O sistema reage à quantidade de carros na rodovia e à capacidade atual da RSU, otimizando o uso de recursos e evitando desperdícios.

Em uma rodovia que possui poucos automóveis, com uma baixa taxa de chegada, a RSU opera normalmente. Diante do aumento do tráfego na rodovia ao longo do dia, a RSU inicialmente operando com baixa capacidade, ajusta-se dinamicamente a esta alta taxa de chegada por meio do auto-escalonamento e aloca novas VMs para ampliar sua capacidade. Por fim, quando a rodovia possui poucos carros, baixando novamente a taxa de chegada, a RSU detecta e por meio do auto-escalonamento, o sistema remove as VMs ociosas para evitar desperdício de recursos.

A Figura 2 apresenta um fluxograma que envolve todos estes estados para auxiliar na compreensão da arquitetura adotada no sistema. O fluxograma inicia com o tráfego

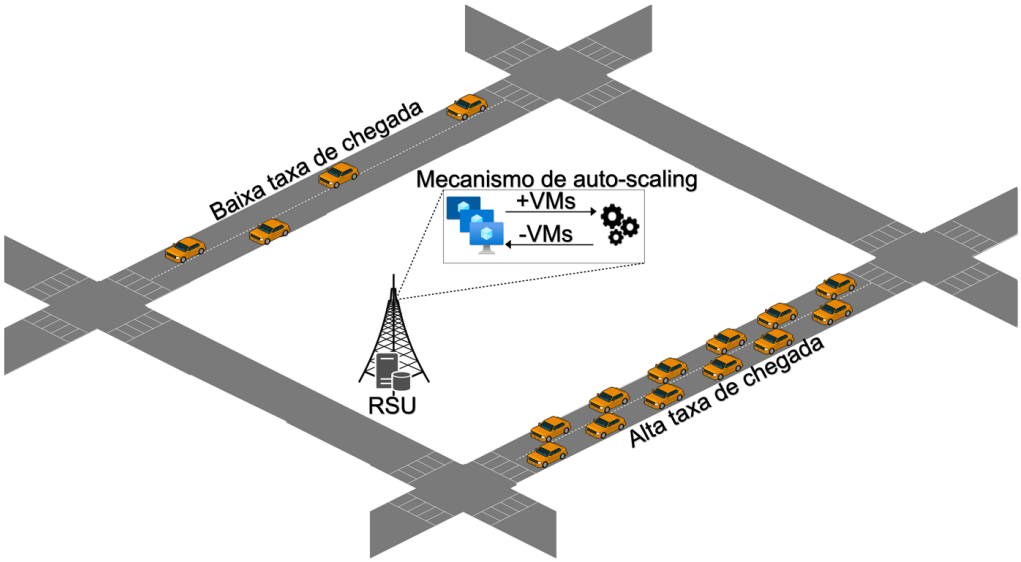


Figura 1. Arquitetura Adotada.

normal de veículos, em que a RSU realiza a captura de dados. Utilizando uma rede wireless para a transmissão eficiente das informações, verifica se há alterações no fluxo de tráfego. Caso não haja alterações, a RSU continua operando normalmente. No entanto, se for detectado um aumento no fluxo de veículos e a fila ficar sobrecarregada, é ativado o primeiro mecanismo de auto escalonamento.

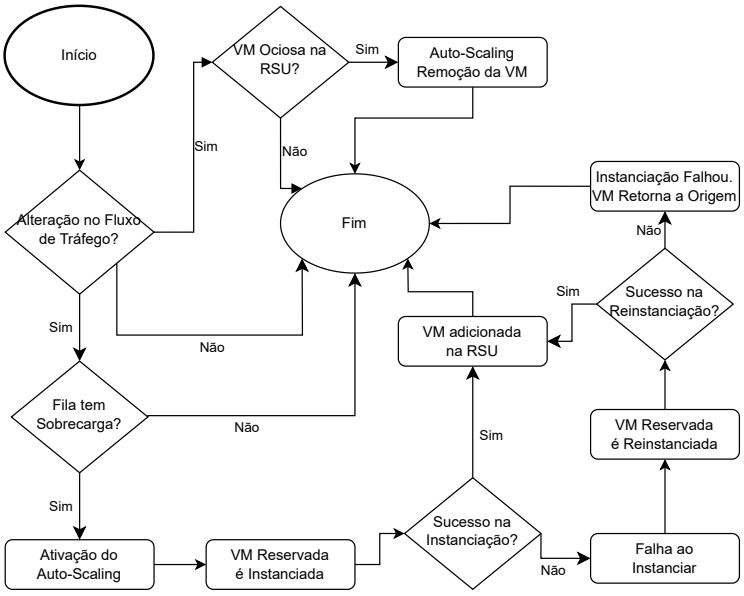


Figura 2. Fluxograma representando o mecanismo de auto-escalonamento.

Com o mecanismo de auto escalonamento ativo, uma nova VM é instanciada. Em caso de sucesso, a VM é adicionada à RSU, que retorna ao seu funcionamento normal. Se ocorrer uma falha ao instanciar esta VM, o sistema de reinstanciação é acionado para garantir processo de escalonamento automático. No segundo momento do mecanismo de auto escalonamento, verifica-se a existência de VMs ociosas. Caso existam, o mecanismo se encarrega de remover essas VMs ociosa, visando à economia de recursos. O resultado

esperado é um sistema adaptável que garanta o monitoramento contínuo da rodovia, com a RSU funcionando normalmente, sendo este o fim do fluxograma. Dessa forma, os recursos são ampliados ou reduzidos conforme a demanda do tráfego, sendo os dados processados utilizados para o monitoramento de veículos e rodovias.

4. Modelo SPN Proposto

Esta seção apresenta o modelo desenvolvido de acordo com as particularidades da arquitetura adotada. A Figura 3 apresenta o modelo base, composto por quatro partes: (1) Admissão, (2) RSU, (3) Mecanismo de *Auto-escalonamento*, e (4) Mecanismo de Reinstanciação.

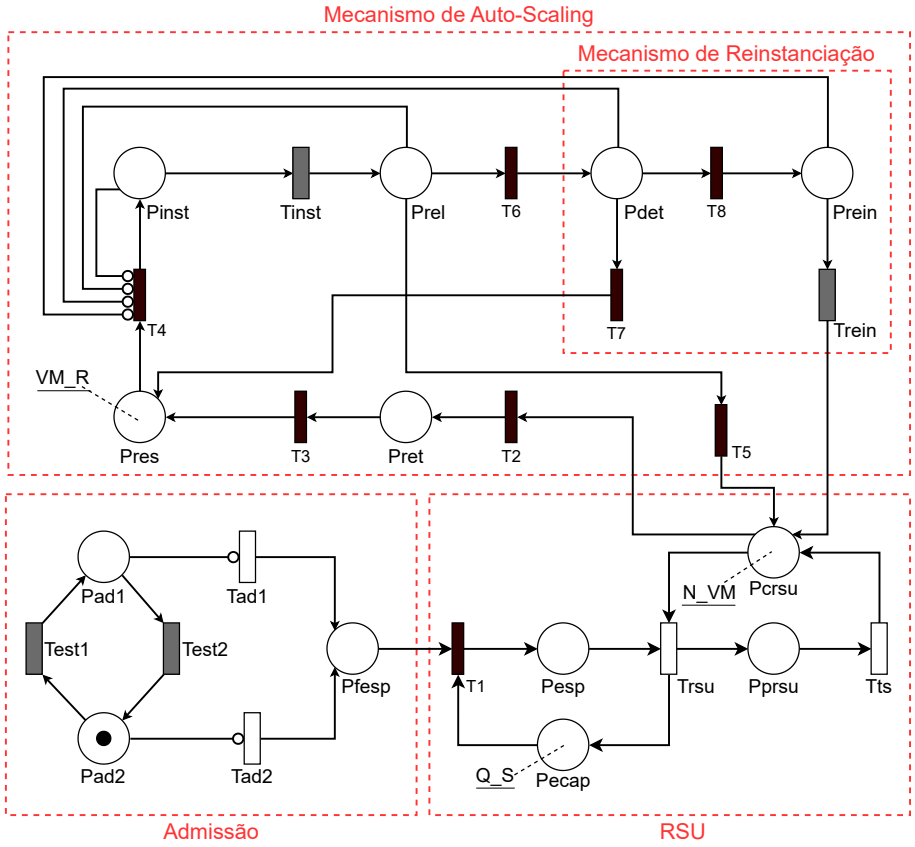


Figura 3. Modelo desenvolvido a partir da arquitetura adotada.

A Admissão é a primeira parte do sistema, responsável pela entrada de requisições no sistema. Ela possui dois lugars de admissão (`Pad1` e `Pad2`), os quais representam diferentes variações no volume de tráfego da rodovia. As duas transições determinísticas (`Test1` e `Test2`), indicam os intervalos de tempo de mudança no volume de tráfego. O lugar `Pfesp` recebe requisições geradas pelos lugars de admissão e habilitam a transição `T1`. O segundo bloco do modelo consiste no módulo da RSU, quando a transição `T1` é disparada. O lugar `Pesp` armazena as requisições que chegam no sistema em uma fila, aguardando o repasse desses dados para o processamento.

O `N_VM` representa a capacidade/quantidade inicial de tokens/VMs disponíveis nessa RSU para processamento. O disparo da transição `Trsu` representa o repasse da requisição para início do processamento da requisição pela VM da RSU, o que ocorre

no lugar (Pprsu). Para que esse processamento seja efetuado, é imperativo que haja VMs disponíveis no lugar Pcrsu, indicando a capacidade da RSU em processar uma requisição. Quando a transição Tts é ativada, a VM que estava processando uma requisição retorna ao lugar de capacidade.

A terceira parte do sistema é o Mecanismo de *Auto-escalonamento*, contemplando as duas etapas do auto escalonamento, que instancia ou libera de capacidade do sistema. A primeira etapa é desencadeada pela transição T4 quando detecta um aumento no enfileiramento e redução na capacidade de processamento da RSU (condição de guarda presente na Tabela 2). Nesta transição, a capacidade reservada de instanciar VMs no lugar Pres (onde VM_R representa o número inicial de máquinas virtuais destinadas ao escalonamento automático) é ativada e encaminhada para o lugar da instância Pinst. Aqui, a transição determinística Tinst representa o tempo de instanciação da VM, o qual seu disparo cria um token no lugar Prel, que a direciona para Pcrsu, o lugar de capacidade RSU, mencionado na segunda parte do sistema, através da transição T5. A segunda etapa do auto escalonamento automático funciona para liberação de recursos quando a RSU tem baixa demanda de processamento. Quando essa diminuição na demanda é identificada, a transição T2 é acionada, coletando as VMs que não estão mais em uso. Essas VMs são direcionadas para o lugar de fallback Pret e, por meio da transição T3, retornam para o lugar inicial em Pres.

A quarto parte do sistema é o Mecanismo de Reinstanciação, cujo objetivo é garantir a entrega bem sucedida da VM ao lugar de capacidade da RSU, em caso de falha durante a instanciação. A transição T6 é ativada, encaminhando a VM para o lugar de detecção do Pdet. Em caso de detecção de falha, a VM retorna ao lugar original através da transição T7. Porém, se a falha for corrigida, a transição T8 é acionada, direcionando a VM para o lugar de reinstanciação Prein. Através da transição determinística Trein, a VM é direcionada para o lugar de capacidade da RSU, Pcrsu. Esta estrutura busca garantir a robustez e continuidade operacional do sistema diante de potenciais falhas durante o processo de instanciação.

4.1. Condições de Guarda e Métricas do Modelo

A Tabela 2 apresenta a implementação das condições de guarda nas transições T2, T4, T5, T7 e T8. As condições de guarda são necessárias para a ativação dos mecanismos de auto escalonamento e reinstanciação. A condição de guarda de T2 é ativada quando se verifica a existência de VMs ociosas em Pcrsu. Ela remove as VMs ociosas, levando-as de volta para o lugar de VMs reservadas pelo sistema (Pres). A condição de T4 é ativada quando Pcrsu possui menos da metade da capacidade inicial de VMs, iniciando o auto escalonamento, no qual uma nova VM será instanciada. A condição de guarda de T5 refere-se à possibilidade de falha. Já T7 e T8 são relacionadas à possibilidade de reinstanciação das VMs.

Para este trabalho, foram obtidas métricas de Tempo Médio de Resposta, Utilização, Probabilidade de Descarte e Vazão. A mensuração do Número de VMs Utilizadas (NVMU) é calculada pela soma do número esperado de tokens de capacidade e em uso presentes em Pcrsu e Pprsu, conforme apresentado na Equação (1).

$$\textbf{NVMU} = (E\{\#Pprsu\}) + (E\{\#Pcrsu\}) \tag{1}$$

Tabela 2. Condições de guarda utilizadas no modelo.

Transição	Condição de Guarda
T2	((#Pcrsu+#Pprsu)>(N_VM*(N_VM/2)))AND(#Pprsu<=(N_VM/(N_VM/2)))
T4	(#Pcrsu<=(N_VM/(N_VM/2)))
T5	P_F
T7	1-P_R
T8	P_R

O Tempo Médio de Resposta do sistema é apresentado na Equação (2). O tempo médio de resposta (*MRT*) pode ser obtido a partir da Lei de Little [Jain 1990]. Esta lei requer um sistema estável, ou seja, que possua uma taxa de requisições menor que a taxa de processamento dos servidores. A lei indica que o Tempo Médio de Resposta (MRT), no nosso caso denominado como o Tempo Médio da Missão (TMM), é dado pela divisão do número médio de requisições em progresso em um sistema ($RequestsInProgress$) pela taxa de chegada de novas entregas (AR). A taxa de chegada é o inverso do tempo entre chegadas. O tempo entre chegadas abreviamos aqui para AD. O valor de $RequestsInProgress$ é dado pela soma de tokens em cada um dos locais que representam uma requisição em andamento. $Enomedolocal$ representa a esperança estatística de existir tokens em "nomedolocal", onde $Enomedolocal = (\sum_{i=1}^{n} P(m(Local) = i) \times i)$, sendo n o maior número de tokens que o $Local$ pode conter. Em outras palavras, $Enomedolocal$ indica o valor esperado de tokens naquele Local. Portanto, MRT do nosso sistema é apresentado na Equação (2). Para calcular essa métrica, leva-se em consideração a soma do número esperado de requisições presentes na fila e em processamento na RSU, multiplicado pelo tempo entre chegadas de novas requisições. Para seguir a Lei de Little se faz necessário "descontar"o descarte de requisições no sistema, ou seja, dividindo o MRT pela probabilidade de tokens não serem descartados.

$$\mathbf{MRT} = \frac{(E\{\#Pesp\} + E\{\#Pprsu\}) * AD}{1 - P\{\#Pecap = 0\}} \tag{2}$$

O nível de utilização de recursos (U) é calculado somando-se a quantidade de recursos utilizados e dividindo pela capacidade total previamente disponível. Já a utilização do sistema é dada pela Equação (3). Fazemos a multiplicação por 100, pois o valor probabilístico do modelo é dado entre 0 e 1 pelo Mercury.

$$\mathbf{U} = \frac{E\{\#Pprsu\}}{N\_VM} * 100 \tag{3}$$

A Equação (4) representa a probabilidade de descarte do sistema (DP). O DP é necessário para avaliar a eficiência de um sistema. Nesse contexto foi utilizada para compreender a capacidade do sistema em lidar com variações na demanda. Existe descarte quando não há mais recursos disponíveis e o lugar responsável por esta informação é (`Pecap`), bastando portando verificar sua nulidade. Na Equação (5) temos a vazão do sistema. A vazão se torna relevante para esse contexto, pois seu resultado reflete a capacidade do sistema em escalar horizontalmente para atender a demanda variável de tráfego. Sendo medida pela esperança de tokens no lugar de processamento da RSU, dividido pelo tempo que a requisição subsequente leva para ser processada (Tts).

$$\textbf{DP} = P\{\#Pecap = 0\} * 100 \tag{4}$$

$$\textbf{TP} = \frac{E\{\#Pprsu\}}{Tts} \tag{5}$$

5. Estudos de Caso

Esta seção apresenta os resultados obtidos como estudos de caso. Os resultados foram obtidos através da ferramenta Mercury (versão 5.0.1) [1] [Silva et al. 2015]. Devido à complexidade da natureza do modelo, os estudos de caso foram conduzidos por meio de simulações estacionárias, incorporando uma margem de erro aproximada de 2%. Os parâmetros utilizados para as simulações são apresentados na Tabela 3.

Tabela 3. Parâmetros utilizados no modelo.

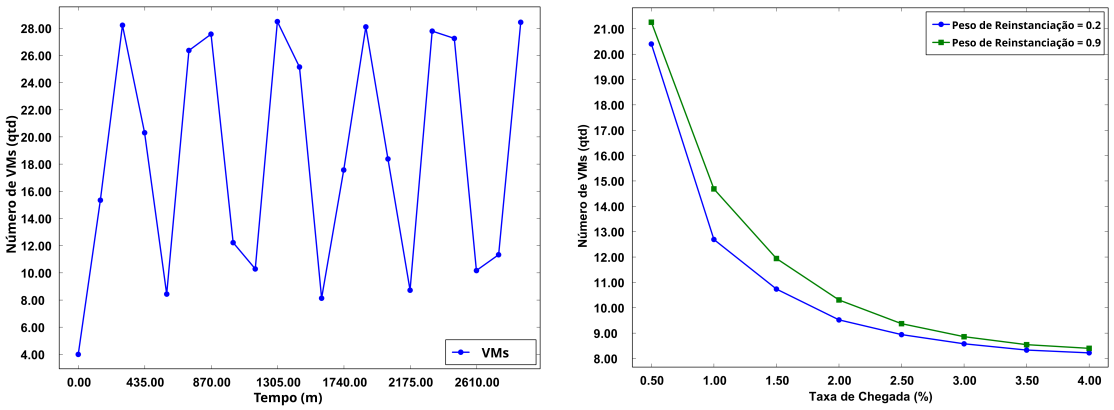
Tipo	Componente	Valor	Definição
Places	Pad1, Pad2	1.0, 0.0	Novas requisições no sistema.
	Pfesp	0.0	Fila de espera das requisições.
	Pesp	0.0	Fila das requisições a processar.
	Pecap	Q_S	Capacidade da fila de requisições.
	Pprsu	0.0	Processamento da RSU.
	Pcrsu	N_VM	Capacidade da RSU.
	Pres	VM_R	VMs reservadas no sistema.
	Pret	0.0	Place de retorno das VMs.
	Pinst	0.0	Place de instanciação de VMs.
	Prel	0.0	Place de liberação das VMs.
	Pdet	0.0	Place de detecção de falha.
	Prein	0.0	Place de reinstanciação
Transições Temporizadas	Test1, Test2	100.0	Transição entre intervalo de tempo do Sistema.
	Tad1, Tad2	AD1, AD2	Tempo de chegada de requisições.
	Trsu	0.1	Transição de chegada na RSU.
	Tts	15.7	Transição de tempo de Serviço.
	Tinst	1.0	Transição de instanciação de VMs.
	Trein	2.0	Transição de reinstanciação de VMs.
Variáveis	AD1, AD2	10.0, 0.7	Chegada dos dados.
	Q_S	1000.0	Tamanho da Fila.
	N_VM	4.0	Número de VMs iniciais da RSU.
	VM_R	25.0	VMs reservadas pelo sistema.
	P_F, P_R	0.1, 0.1	Peso de Falha e Peso de Reinstanciar.

Os valores atribuídos aos parâmetros dos componentes do sistema foram derivados de fontes acadêmicas previamente publicadas, destacando-se especialmente nos trabalhos [Fé et al. 2022b, Carvalho et al. 2020]. As duas subseções seguintes destacam o potencial uso do modelo.

[1] https://www.modcs.org/

5.1. Estudo de Caso 1 - Análise de Adaptabilidade do Sistema

Esta subseção apresenta os resultados de adaptabilidade obtidos por meio da análise do sistema. Um dos pontos centrais do trabalho foi assegurar a disponibilidade eficiente do sistema durante os picos de tráfego nos diferentes períodos do dia. Isso, aliado à prevenção de desperdício de recursos, por meio do mecanismo de auto escalonamento, incluindo a redução de VMs ociosas durante a noite, por exemplo. Além disso, para economizar recursos, foi medida a taxa de reinstanciação de VMs, buscando garantir que o processo de instanciação fosse repetido caso houvesse alguma falha ao encaminhar uma VM para a RSU.



(a) Número de VMs em Uso (NVMU). (b) Reinstanciação de VMs no Sistema.

Figura 4. Adaptabilidade do sistema a variação do número de VMs.

A Figura 4 apresenta tanto a utilização da adaptabilidade através da variação da demanda de VMs ao longo do dia quanto o processo de reinstanciação em caso de falha. A Figura 4(a) apresenta a variância no uso de VMs ao decorrer do dia, denominamos esta métrica de NVMU (número de VMs em uso). Sua adaptabilidade depende da troca de estados utilizada no modelo, inicialmente com 4 VMs. Para que a análise fosse viável, delimitou-se um período de 2 dias, totalizando 2880 minutos, tempo no eixo x. Na análise base havia 25 VMs reservadas disponíveis, mais as 4 iniciais na RSU. Ao decorrer do primeiro espaço de tempo, chega a utilizar todas as VMs reservadas disponíveis. Após isso, quando o tráfego de carros na rodovia começa a diminuir, o sistema detecta que essa quantidade de VMs não é mais necessária. Isto se dá para evitar o desperdício de recursos, podendo encaminhar essas VMs para uma outra RSU, por exemplo. Assim, levando a linha do gráfico a um comportamento de declínio.

Para que esta análise fosse possível, utilizou-se um método de Análise Transiente. Este método analítico oferece uma visão detalhada e dinâmica da evolução temporal do sistema, proporcionando informações essenciais sobre sua capacidade de adaptação a flutuações na demanda ao longo do tempo. Por sua dinamicidade, pode-se detectar comportamentos transitórios, como picos de carga. Isso foi possível por meio de estados variáveis do modelo (Test1, Test2 da Tabela 3).

Já a Figura 4(b) representa os resultados do mecanismo de reinstanciação do sistema. Esse mecanismo buscou diminuir a falha ao instanciar novas máquinas virtuais para a RSU. A partir do aumento da porcentagem da taxa de chegada (eixo x do gráfico),

o eixo y do gráfico representa o número de VMs reservadas. Essa análise iniciou com um número acima de 21.0 Máquinas Virtuais. É notável que a partir do início do uso do sistema, com a taxa de chegada ainda menor que 1.0%, já inicia-se a instanciação das VMs.

Ao decorrer do aumento da porcentagem da taxa de chegada, o número de VMs tem comportamento de queda, ou seja, as VMs estão sendo encaminhadas para a RSU por meio do auto-escalonamento. Com isso, é demonstrado que o comportamento do mecanismo de reinstanciação é o esperado, já que existe um reação de queda continua. Caso o mecanismo de reinstanciação falhasse, as linhas deveriam ter comportamento crescente, pois, as VMs reservadas estariam retornando para sua origem.

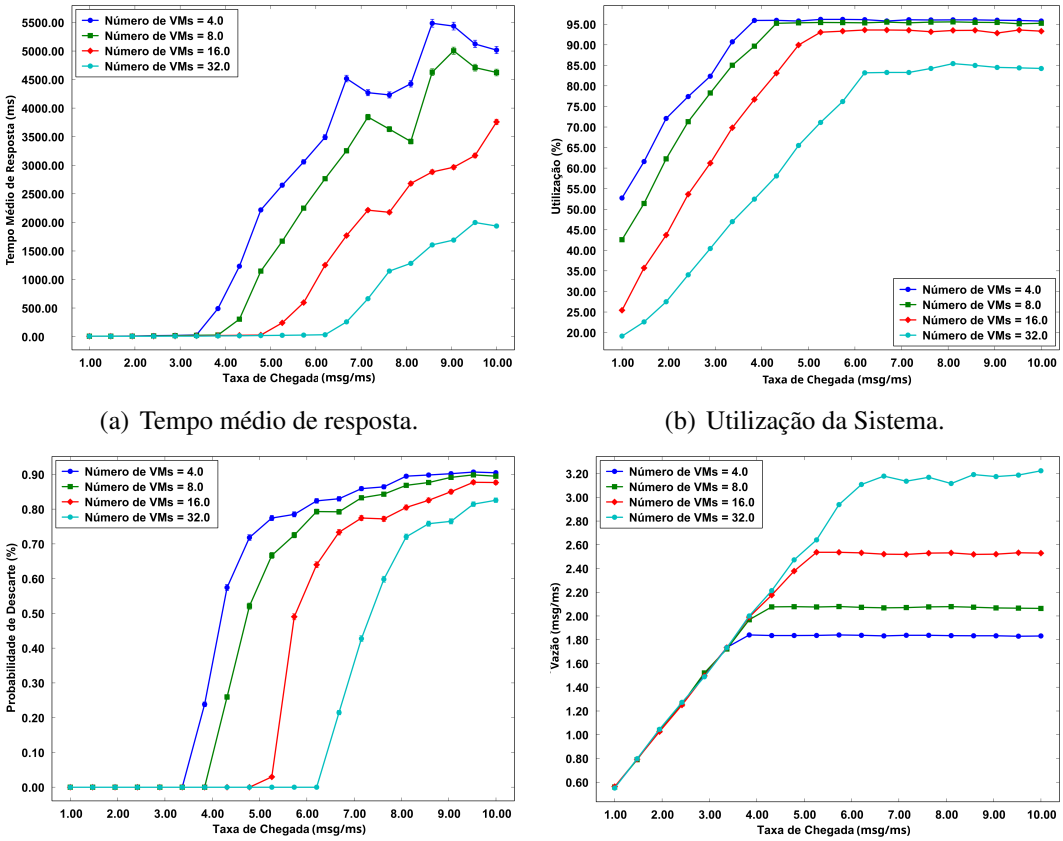
5.2. Estudo de Caso 2 - Análise de Desempenho

Nesta subseção, conduz-se uma análise dos resultados de desempenho decorrentes da variação da quantidade inicial de VMs no sistema. Observou-se que, ao longo da análise, o número de VMs surge como um dos fatores preponderantes tanto no tempo de resposta do sistema quanto na sua utilização efetiva. Para uma compreensão mais abrangente dos resultados, os gráficos correspondentes à probabilidade de descarte e à vazão do sistema também são incorporados nesta avaliação.

A Figura 5 apresenta os resultados da análise que aborda a variação do número inicial de VMs. Essa variação é em relação ao Tempo Médio de Resposta (MRT), Utilização do sistema, Probabilidade de Descarte (Drop Probability) e a Vazão do Sistema (Throughput). A Figura 5(a) apresenta um aumento significativo no Tempo Médio de Resposta à medida que a taxa de chegada aumenta. É fundamental destacar que a redução do número inicial de VMs resulta em um maior MRT no sistema. Por exemplo, ao empregar 4 VMs (menor valor inicial utilizado), o sistema mantém desempenho satisfatório até 4,00 mensagens por milissegundo, a partir das quais o MRT aumenta progressivamente, atingindo aproximadamente 5.500,00 ms. Os demais resultados indicam uma tendência de aumento, porém, ao aumentar o número de VMs iniciais, esse aumento não é tão pronunciado como no caso do menor valor inicial utilizado. Tomando como exemplo o valor mais alto, 32 máquinas virtuais iniciais começam a crescer a partir de aproximadamente 6,00 mensagens, com o MRT atingindo cerca de 2.000,00 ms.

A Figura 5(b) dispõe a porcentagem de utilização do sistema a partir da variação do número inicial de VMs. Pode-se observar que ao iniciar o sistema inicia com mais de 50% de utilização com 4 VMs, o mesmo tem chega ao seu pico máximo de 95% antes mesmo de 4.00 msg/ms. Seguido da segunda variação com 8 VMs, que também chega ao uso máximo do sistema, acima de 95%, porém, a partir de 4.00 msg/ms. O maior número de VMs utilizado no sistema não chegou ao pico máximo, ficando aproximada em 85% de utilização a partir de 6.00 msg/ms. Uma alta utilização, como nas duas primeiras variações, acarreta em uma maior taxa de descarte, considerando assim que o sistema está em um estado estressado.

A Figura 5(c) demonstra a probabilidade de descarte do sistema. Os resultados gerais mostram que o sistema é estressado até cerca de no máximo 90% com 4.0 e 8.0 VMs, mais de 85% com 16.0 VMs e aproximadamente 80% com 32 VMs. O gráfico de descarte, aqui apresentado, desempenha um papel complementar MRT. Analisando a relação entre ambos, observa-se que o MRT inicia seu crescimento, atingindo posterior-



(a) Tempo médio de resposta.

(b) Utilização da Sistema.

(c) Probabilidade de Descarte.

(d) Vazão do Sistema.

Figura 5. Impacto da variação do número de máquinas virtuais no desempenho do sistema.

mente um ponto de inflexão em que inicia uma diminuição. Esse comportamento sugere uma dinâmica interessante no sistema, onde o descarte de mensagens se relaciona de maneira inversa, com MRT crescer, a probabilidade de descarte influencia na eficiência operacional do sistema.

O vazão, representado na Figura 5(d), em função da taxa de chegada para diferentes números de Máquinas Virtuais (VMs), ilustra a capacidade do sistema de processar mensagens em um determinado intervalo de tempo. Com 4 VMs, o fluxo atinge seu pico, aproximadamente 1,80 a 2,0 msg/ms no eixo y, antes de atingir uma estabilização em torno de 4,00 no eixo x. Por outro lado, através do auto-escalonamento por exemplo, ao usar 32 VMs a taxa de transferência experimenta um crescimento contínuo até aproximadamente 3,20 no eixo y, estabilizando em 6,0 no eixo x. Com isso, a vazão reflete a eficiência do sistema ao lidar com a carga de trabalho, revelando que o ajuste adequado do número de máquinas virtuais.

6. Conclusão

Este artigo propôs um modelo de redes de SPN para uma arquitetura de monitoramento de tráfego VANET com variabilidade temporal, que altera a densidade do tráfego. Devido a essa variação significativa, implementou-se um mecanismo de auto escalonamento com o objetivo de manter o sistema disponível e aprimorar o desempenho, mesmo diante de

flutuações no fluxo de carros em uma determinada rodovia. O modelo contempla uma análise abrangente de diversos fatores que potencialmente influenciam o desempenho do sistema. Para analisar o modelo foram utilizadas as métricas tempo médio de resposta, utilização, probabilidade de descarte e vazão do sistema. Além disso, foram consideradas métricas para a comprovação que o mecanismo auto escalonamento e o mecanismo de reinstanciação seriam eficientes. Os resultados mostram que a quantidade de VMs iniciais no sistema são elementos de configuração importantes no planejamento. No geral, os resultados mostram que os mecanismos auto escalonamento e de reinstanciação melhoraram o desempenho final do sistema. Como trabalhos futuros, pretende-se realizar uma análise de sensibilidade para verificar quais outros fatores são essenciais para esse tipo de arquitetura. Além disso, realizar um experimento de validação completo para o modelo, melhoria na arquitetura e adicionar mais mecanismos que promovam a melhora no desempenho do sistema.

Referências

Araújo, G., Rodrigues, L., Oliveira, K., Fé, I., Khan, R., and Silva, F. A. (2021). Vehicular cloud computing networks: Availability modelling and sensitivity analysis. *International Journal of Sensor Networks*, 36(3):125–138.

Bobbio, A., Puliafito, A., Telek, M., and Trivedi, K. S. (1998). Recent developments in non-markovian stochastic petri nets. *Journal of Circuits, Systems, and Computers*, 8(01):119–158.

Carvalho, D., Rodrigues, L., Endo, P. T., Kosta, S., and Silva, F. A. (2020). Edge servers placement in mobile edge computing using stochastic petri nets. *International Journal of Computational Science and Engineering*, 23(4):352–366.

Cumbal, R., Gutiérrez, S., Guerrero, C., Hincapié, R., and Arévalo, G. (2019). Optimal resources allocation from vanet infrastructures in dynamic mobile environments. In *2019 IEEE Latin-American Conference on Communications (LATINCOM)*, pages 1–5. IEEE.

Fé, I., Matos, R., Dantas, J., Melo, C., Nguyen, T. A., Min, D., Choi, E., Silva, F. A., and Maciel, P. R. M. (2022a). Performance-cost trade-off in auto-scaling mechanisms for cloud computing. *Sensors*, 22(3):1221.

Fé, I., Matos, R., Dantas, J., Melo, C., Nguyen, T. A., Min, D., Choi, E., Silva, F. A., and Maciel, P. R. M. (2022b). Performance-cost trade-off in auto-scaling mechanisms for cloud computing. *Sensors*, 22(3):1221.

Jain, R. (1990). *The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling*. John Wiley & Sons.

Karabulut, M. A., Shah, A. S., Ilhan, H., Pathan, A.-S. K., and Atiquzzaman, M. (2023). Inspecting vanet with various critical aspects–a systematic review. *Ad Hoc Networks*, page 103281.

Macêdo, J., Carvalho, V., Andrade, E., and Silva, F. (2022). Modeling and analysis of communication in vanets using rsus. In *Proceedings of the 21st Workshop on Performance of Computer and Communication Systems*, pages 96–107, Porto Alegre, RS, Brasil. SBC.

Martin-Faus, I. V., Urquiza-Aguiar, L., Aguilar Igartua, M., and Guérin-Lassous, I. (2018). Transient analysis of idle time in vanets using markov-reward models. *IEEE Transactions on Vehicular Technology*, 67(4):2833–2847.

Naresh, R., Narayanan, K. L., Kumar, C. V., and Senthilkumar, S. (2024). A routing in vanet towards smart business cities using optimization techniques. In *Digital Twin Technology and AI Implementations in Future-Focused Businesses*, pages 1–13. IGI Global.

Ouhmidou, H., Nabou, A., Ikidid, A., Bouassaba, W., Ouzzif, M., and El Kiram, M. A. (2023). Traffic control, congestion management and smart parking through vanet, ml, and iot: A review. In *2023 10th International Conference on Wireless Networks and Mobile Communications (WINCOM)*, pages 1–6. IEEE.

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.

Rodrigues, L., Neto, F., Gonçalves, G., Soares, A., and Silva, F. A. (2021). Performance evaluation of smart cooperative traffic lights in vanets. *International Journal of Computational Science and Engineering*, 24(3):276–289.

Siddiqi, M. H., Alruwaili, M., Ali, A., Haider, S. F., Ali, F., and Iqbal, M. (2020). Dynamic priority-based efficient resource allocation and computing framework for vehicular multimedia cloud computing. *IEEE access*, 8:81080–81089.

Silva, B., Matos, R., Callou, G., Figueiredo, J., Oliveira, D., Ferreira, J., Dantas, J., Lobo, A., Alves, V., and Maciel, P. (2015). Mercury: An integrated environment for performance and dependability evaluation of general systems. In *Proceedings of the Industrial Track at 45th Dependable Systems and Networks Conference, DSN*, pages 1–4.

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.

Tang, Y., Cheng, N., Wu, W., Wang, M., Dai, Y., and Shen, X. (2019). Delay-minimization routing for heterogeneous vanets with machine learning based mobility prediction. *IEEE Transactions on Vehicular Technology*, 68(4):3967–3979.

Wu, X., Zhao, S., Zhang, R., and Yang, L. (2020). Mobility prediction-based joint task assignment and resource allocation in vehicular fog computing. In *2020 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6. IEEE.