

Um Novo Mecanismo para Alocação de Recursos em Clouds Veiculares

Adinovam H. M. Pimenta¹, Azzedine Boukerche², Rodolfo I. Meneguette^{2,3}

¹ Universidade Federal de São Carlos - UFSCAR

² School of Electrical Engineering and Computer Science - EECS

³ Instituto Federal de São Paulo - IFSP

adinovam_pimenta@dc.ufscar.br, {boukerch, rmenegue}@site.uottawa.com,

Abstract. *Intelligent Transportation Systems are increasingly attracting attention not only from academics but also industries. These systems aim to improve the quality of transportation through monitoring, management of the transportation system, as well as make the trip more comfortable and safer for drivers and passengers. Intelligent transportation system can benefit from the use of mobile clouds. Due to the characteristics of the vehicular networks, the management, and allocation of the resources of this cloud becomes a challenge. Thus, we propose a mechanism of allocation and management of resources independent of any external infrastructure. Therefore, the mechanism does not consider the communication between vehicles with a roadside unit infrastructure. For this, we formulate the problem of allocation of resources through the Semi-Markov Decision Process (SMDP), which will provide an optimal solution to the problem. In addition, we also utilize the function of reward and an interactive algorithm. The results obtained through simulation showed that the proposed mechanism has a stable behavior independent of the frequency of receiving the service requests.*

Resumo. *Sistemas de Transporte Inteligente vêm chamando cada vez mais a atenção não só da academia mas também da indústria. Esses sistemas têm como objetivo melhorar a qualidade do transporte através de aplicações de monitoramento, gerenciamento do sistema de transporte, bem como tornar a viagem mais confortável e segura para os condutores e passageiros. Sistema de transporte inteligente pode se beneficiar do uso de nuvens móveis. Devido as características das redes veiculares, torna-se um desafio o gerenciamento e alocação dos recursos dessa cloud. Assim, foi proposto neste trabalho um mecanismo de alocação e gerenciamento de recursos independente de qualquer infraestrutura externa. Portanto, o mecanismo não considera a comunicação entre o veículo com uma infraestrutura de acostamento. Para isso, formulamos o problema de alocação de recurso através do Semi-Markov Decision Process (SMDP) que fornecerá uma solução ótima para o problema. Além disso, também utilizamos uma função da recompensa e de um algoritmo iterativo. Os resultados obtidos através de simulação mostraram que o mecanismo proposto possui um comportamento estável independente da frequência do recebimento das requisições de serviço.*

1. Introdução

Um sistema de transporte inteligente consiste de um conjunto de tecnologias aplicadas ao melhoramento da qualidade dos sistemas de transporte [Liu et al. 2016], provendo ser-

viços e recursos com o objetivo de tornar a viagem do condutor e dos passageiros dos veículos mais confortável e segura [Sladkowski and Pamula 2015, Ghazal et al. 2015]. Esses sistemas são disponibilizados para os usuários por meio de uma infraestrutura baseada nas redes veiculares [Ashokkumar et al. 2015]. Portanto, esses sistemas utilizam da comunicação dos veículos com outros veículos e da comunicação dos veículos com uma infraestrutura de acostamento para oferecer serviços aos seus usuários [Meneguette 2016]. Um sistema de transporte inteligente também pode utilizar os dispositivos e sensores embarcados no veículo para aumentar a sua quantidade de recursos [Cheng et al. 2015, Khan et al. 2016]. Para isso, tais sistemas utilizam do sistema de *clouds* veiculares para agregar recursos dos veículos e oferecer serviços aos seus usuários independentemente da localização do veículo [Chirayil and Thomas 2016].

Uma *Clouds* veicular é um grupo de veículos que cooperativamente detectam, comunicam, computam e disponibilizam seus recursos físicos dinamicamente através da cooperação entre os veículos e veículos com uma unidade de acostamento [Ghazizadeh et al. 2016]. Essas *clouds* veiculares podem usar tanto os veículos quanto uma infraestrutura externa para coordenar a alocação e o gerenciamento das requisições de recursos e serviços [Farooq et al. 2016]. Na literatura existem alguns trabalhos direcionados à alocação de recursos nas *clouds* veiculares [Zheng et al. 2015, Meng et al. 2015, Arkian et al. 2015, Miao et al. 2016, Dai et al. 2015, Pyun et al. 2016]. Nos trabalhos de [Zheng et al. 2015, Meng et al. 2015] os autores utilizaram o *Semi-Markov Decision Process* (SMDP) para realizar a alocação de recursos em uma *cloud* veicular. Já nos trabalhos [Arkian et al. 2015, Miao et al. 2016] os autores utilizaram-se da lógica *fuzzy* para auxiliar no processo de alocação dos recursos bem como na seleção dos veículos que iria utilizar na comunicação entre diferentes *clusters*. Nos trabalhos [Dai et al. 2015, Pyun et al. 2016] os autores utilizaram modelos matemáticos para lidar com o problema da alocação em redes veiculares. Para isso, os autores utilizam a comunicação direta entre o veículo e a infraestrutura de acostamento. No trabalho de [Yu et al. 2015] os autores utilizam teoria de jogos para auxiliar no processo de alocação de recursos. Todos esses trabalhos utilizam uma infraestrutura de acostamento para a realização da alocação dos recursos na *cloud* veicular. Além disso, boa parte desses trabalhos não considera que o veículo possui recursos que podem ser agregados na *cloud*, bem como a mobilidade do veículo no momento da alocação do serviço. Portanto, essas soluções não lidam com todos os desafios encontrados em uma rede veicular.

Posto isso, um dos grandes desafios é a realização da alocação e gerenciamento dos recursos pelos próprios veículos [Meneguette et al. 2016]. Assim, os veículos têm que coordenar e cooperar entre si para criar dinamicamente uma infraestrutura de comunicação e controle com intuito de controlar as chegadas de requisições e gerenciar os recursos disponibilizados pelos veículos com objetivo de atender a demanda dos usuários. Com esse desafio em mente, nós propomos um novo mecanismo de alocação de recurso somente entre os veículos, com o objetivo de maximizar a disponibilidade do recurso na *cloud* veicular. Além disso, consideramos que os veículos possuem recursos disponíveis, ou seja, assumimos que o veículo possui vários recursos embarcados que podem ser utilizados pela *cloud*. Para a criação da *cloud* consideramos que os veículos devem cooperar e colaborar entre si para criar uma estrutura que permita o gerenciamento de recurso. Essa estrutura criada é baseada em *cluster*, na qual o *Cluster Head* irá gerenciar as de-

mandas e os recursos disponíveis da sua *cloud*. Portanto, esse *Cluster Head* assumirá o papel de controlador da *cloud* e irá decidir se deve alocar ou não os recursos na *cloud* para atender ao usuário. O mecanismo de alocação de recurso proposto foi formulado através do SMDP que foi resolvido através de um algoritmo de iteração a fim de obter a política ótima. O objetivo é maximizar a recompensa média do sistema e melhorar a disponibilidade dos recursos na *cloud*.

O restante deste artigo está organizado da seguinte forma. Na próxima seção, mostramos uma visão geral das principais abordagens existentes para a alocação de recursos em *clouds* veiculares utilizando SMDP. A nossa proposta de um mecanismo de alocação de recurso é descrita nas seções 3 e 4, enquanto que uma avaliação detalhada do desempenho e dos resultados da simulação são apresentados na seção 5. Finalmente, a seção 6 apresenta as conclusões.

2. Trabalhos Relacionados

Na literatura existem alguns trabalhos que lidam com o problema de alocação de recurso em *clouds* veiculares utilizando SMDP.

Zheng et al. [Zheng et al. 2015] propuseram um mecanismo de alocação de recursos que maximiza a recompensa do sistema de uma *cloud* veicular. Esse mecanismo é baseado no *Infinite Horizon Semi-Markov Process*, que usa: (i) um conjunto de estados que representa a quantidade de recursos atuais e os estados dos serviços requisitados para a *cloud* veicular; (ii) conjunto de ações que podem ser usadas pelo controlador da *cloud* baseado no seu estado atual; (iii) função de recompensa que calcula o ganho e custo de uma tomada de decisão; e (iv) a probabilidade de transição, que calcula a mudança de um estado do sistema para outro estado decorrente de uma ação específica. Esse mecanismo também utiliza um algoritmo iterativo para resolver problema de otimização.

Meng et al. [Meng et al. 2015] propuseram uma melhoria dos trabalhos [Zheng et al. 2015]. Nessa melhoria os autores buscam maximizar o ganho do sistema e melhorar a qualidade de experiência do usuário (veículo). Para isso, os autores também utilizaram o SMDP para otimizar o processo de tomada de decisão do sistema com o objetivo de obter a solução ótima. Para obter essa solução, a recompensa é calculada usando a média da taxa de eventos e o SMDP é passado por um processo de uniformização transformando o modelo em um modelo de tempo discreto. Esse mecanismo também usa um algoritmo iterativo para resolver o SMDP. Diferentemente do trabalho de Zheng [Zheng et al. 2015] esse trabalho tenta melhorar o poder de processamento para economizar mais energia.

Arkian et al. [Arkian et al. 2015] propuseram uma arquitetura para *cloud* veicular, conhecida como COHORT. Essa arquitetura é baseada no mecanismo de aprendizado de máquina. Essa arquitetura utiliza mecanismos como o *Q-learning*, a lógica *fuzzy* e estratégia de clusterização a fim de melhorar e auxiliar o mecanismo de gerenciamento de recursos. A lógica *fuzzy* calcula o FitFactor, que indica qual veículo poderia se tornar um *Cluster Head*. A lógica *fuzzy* usa como entrada informações dos veículos, tais como velocidade, grau de vizinhança, qualidade da conexão entre o veículo e a *roadside unit* (RSU). Com objetivo de auxiliar o gerenciamento de recurso, o COHORT precisa saber quais serão os veículos que irão auxiliar o *Cluster Head* na comunicação tanto interna como externa de seu *cluster*. Essa seleção utiliza os mecanismos de aprendizado por

reforço através do *Q-learning* para selecionar quais serão os melhores veículos que irão auxiliar o *Cluster Head* no gerenciamento de recurso.

A solução proposta neste trabalho difere-se das outras, pois considera que os veículos possuem recursos que podem ser disponibilizados para *cloud* veicular dinâmica aumentando, assim, a capacidade da *cloud* em oferecer mais serviços aos seus usuários (veículos). A solução proposta neste trabalho busca maximizar a disponibilidade dos recursos disponíveis nos veículos com objetivo de maximizar a quantidade de serviços oferecido pela *cloud* veicular.

3. Um Novo Mecanismo para Alocação de Recursos em Clouds Veiculares

Nessa seção descreveremos o modelo de alocação e gerenciamento de requisições de serviços propostos para *clouds* veiculares, chamado de MAVeRIC (Mecanismo de Alocação de recursos em *clouds* VEICulares), que tem como objetivo maximizar a disponibilidade dos recursos na *cloud*. O MAVeRIC não utiliza nenhuma infraestrutura externa para auxiliar na comunicação entre os veículos, bem como na alocação dos recursos. Portanto, os veículos precisam cooperar e colaborar entre si para formarem uma estrutura que permita a disponibilidade de serviços e recursos para os usuários (veículos) da *cloud*. Essa estrutura criará um conjunto de *cluster* na qual permitirá que os veículos possam compartilhar seus recursos, aumentando assim a capacidade da *cloud* veicular.

A formação do *cluster* é baseada no trabalho de Meneguette e colaboradores [Meneguette et al. 2016], que cria um *cluster* entre os veículos sem uma infraestrutura externa. Além disso, os controladores (*Cluster Head*) gerenciam a busca de recursos através de um mecanismo dinâmico de busca. Embora o *Cluster Head* tenha o controle do seu *cluster* (*cloud*), algumas requisições são tratadas diretamente pelo veículo, ou seja, a requisição não passa pelo *Cluster Head*. Neste artigo, consideramos que os veículos têm recursos a serem compartilhados na *cloud*. Além disso, também consideramos que todas as requisições de serviços devem passar pelo *Cluster Head*. Assim, o *Cluster Head* pode ter o controle do estado do *cluster* como um todo, facilitando o gerenciamento e alocação dos recursos. Portanto, dando o poder de decisão de alocação de um determinado recurso ou serviço solicitado. Para isso, formulamos o mecanismo de alocação de recursos usando o SMDP e um algoritmo iterativo para obtenção da solução do problema de alocação. A Figura 1 descreve uma abstração da *cloud* veicular.

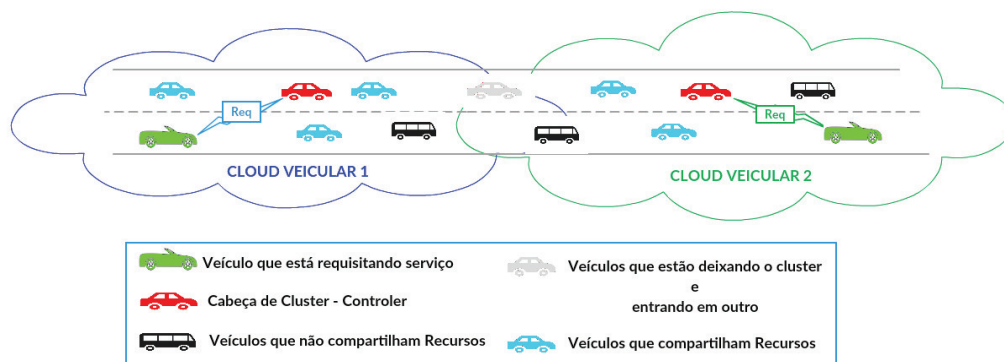


Figura 1. Abstração do MAVeRIC considerando a formação da *cloud* veicular através de *cluster*

4. Formulação do Problema

Para formular o SMDP, assumimos que a entrada dos veículos na *cloud* seguem uma distribuição de Poisson, representada neste trabalho pela variável $\lambda_{chegada}$ e a saída desses veículos da *cloud* segue uma probabilidade representada aqui pela variável $Deixar_{cloud}$. A *cloud* consiste de V veículos que compartilham os recursos entre si. A *cloud* disponibiliza K serviços. Cada serviço precisa de b_k recursos, $k \in \{1, 2, \dots, K\}$, para ser executado. O tempo de chegada de uma requisição de serviço segue também uma distribuição de Poisson com uma taxa λ_k . Cada serviço tem uma duração de μ_k segundos.

Na próxima seção, definimos os elementos que compõem o SMDP utilizado neste trabalho, conjunto de estados, ações, probabilidade de transição e função de recompensa do sistema. A Tabela 1 descreve as notações importantes do modelo.

Tabela 1. Lista de Notações Importantes

Termo	Descrição
$\lambda_{chegada}$	Taxa de chegada do veículo
$Deixar_{cloud}$	Probabilidade da saída do veículo
V	Número de veículos na <i>cloud</i>
k	Número de serviços
b_k	Número de recursos alocados para o serviço k
λ_k	Taxa de chegada de requisições de serviço
μ_k	Duração do serviço
n^{Req}	Número de requisições aceitas
n^{Res}	Número de veículos que estão compartilhando seus recursos
$Cheg_k^{Req}$	Chegada de uma nova requisição de serviço k
$Cheg^{Veículo}$	Chegada de um novo veículo na <i>cloud</i>
Fin_k^{Req}	Término de execução de um serviço k
$Deixa_k^{Veículo}$	Saída de um veículo que disponibilizou seus recursos para o serviço k
$R_{chegada}$	Ganho do sistema com a chegada de um veículo
R_{Req}	Ganho do sistema pelo aceite da requisição
R_{Ter}	Ganho do sistema pelo término do serviço
P_{Deixa}	Penalização do sistema devido à saída de um veículo
P_{Req}	Penalização do sistema devido ao bloqueio do serviço
w_u	Peso da utilização do serviço
w_a	Peso da disponibilidade do serviço

4.1. Conjunto de Estados do Sistema

O conjunto de estados do sistema consiste no número de requisições atendidas pelo sistema; número de veículos que estão compartilhando seus recursos; e um conjunto de eventos que podem ocorrer no sistema. Assim, o conjunto de estado do sistema é formulado como:

$$S = \{s | s = (n^{Req}, n^{Res}, e)\}, \quad (1)$$

Onde n^{Req} é definido como:

$$n^{Req} = \{n_1^{Req}, n_2^{Req}, \dots, n_k^{Req}\} \quad (2)$$

A variável n_1^{Req} descreve a quantidade de veículos na *cloud* que tiveram suas requisições de serviços da classe k atendidas. A variável n^{Res} descreve a quantidade de veículos na *cloud* que estão compartilhando seus recursos. Assim, o total de recursos alocados em uma *cloud* deve obedecer a seguinte restrição: $\sum_{k=1}^K b_k n_k^{Req} \leq n^{Res}$. A variável e representa um evento que pode ocorrer no MAVeRIC, sendo que:

$$e \in E = \{Cheg, Fim_k^{Req}, Deixa_k^{Veiculo}\} \quad (3)$$

Assim, o conjunto de eventos E é definido como:

- $Cheg = \{Cheg^{Veiculo}, Cheg_k^{Req}\}$ na qual $Cheg^{Veiculo}$ representa a chegada de um novo veículo na *cloud* veicular que irá compartilhar seus recursos, e a variável $Cheg_k^{Req}$ representa a chegada de uma nova requisição do serviço k ;
- Fim_k^{Req} representa o término da execução de um serviço da classe k ;
- $Deixa_k^{Veiculo}$ representa a probabilidade de um veículo, cujos recursos foram alocados para a classe k , deixar a *cloud* veicular.

4.2. Conjunto de Ações

A cada período de tempo, o MAVeRIC precisa decidir se irá aceitar ou bloquear uma determinada requisição de serviço. Essas decisões são tomadas com base no conjunto de ações do sistema. Portanto, a cada período de tempo, o MAVeRIC selecionará uma ação a de um conjunto de ações A_s , que é descrito pela equação 4.

$$A_s = \begin{cases} \{0, 1, 2, \dots, K\}, & e \in Cheg_k^{Req} \\ \{-1\}, & e \in \{Cheg^{Veiculo}, Fim_k^{Req}, Deixa_k^{Veiculo}\} \end{cases} \quad (4)$$

A ação $a \in \{1, 2, \dots, K\}$ representa a aceitação da requisição do serviço da classe k , ou seja, o sistema alocará os recursos necessários para atender a essa requisição. Quando $a = 0$ indica que houve um bloqueio da requisição do serviço da classe k , ou seja, o sistema irá rejeitar essa requisição. Quando $a = -1$ indica que somente será realizada a atualização das informações do sistema, portanto, o MAVeRIC não irá tomar nenhuma decisão.

4.3. Probabilidade de Transição

A probabilidade de transição entre um estado s para um estado s' decorrente de uma ação a é definida por $p(s'|s, a)$

$$p(s'|s, a) = \begin{cases} \frac{\lambda_{classes_k}}{\varphi(s', a)} & e' = Cheg_k^{Req} \\ \frac{\lambda_{chegada}}{\varphi(s', a)} & e' = Cheg^{Veiculo} \\ \frac{\mu_k n_k^L}{\varphi(s', a)} & e' = Fim_k^{Req} \\ \frac{Deixar_{cloud} * n^{res}}{\varphi(s', a)} & e' = Deixa_k^{Veiculo} \end{cases} \quad (5)$$

Onde $\varphi(s, a) = \gamma(s, a)^{-1}$

$$\gamma(s, a) = \frac{1}{\lambda_{classes_k} + \lambda_{chegada} + 2 * n^{res} * Deixar_{cloud} + \sum_k^K \mu_k * n_k^{Req}} \quad (6)$$

Onde $\lambda_{classes_k} = \sum_k^K \lambda_k * \left\lfloor \frac{n^{Res}}{b_k} \right\rfloor$.

4.4. Função de Recompensa

Considerando um estado s e uma ação a , podemos definir a função de recompensa como:

$$r(s, a) = p(s, a) - g(s, a), \quad (7)$$

Onde $p(s, a)$ é o ganho fixo do sistema após a tomada de uma ação a . A função $g(s, a)$ define o custo esperado do sistema, representando o custo por unidade de tempo durante o período do serviço. A função $p(s, a)$ é expressa como:

$$p(s, a) = \begin{cases} P_{Req} & e \in Cheg_k^{Req}, a = 0 \\ R_{Req} & e \in Cheg_k^{Req}, a \in \{1, 2, \dots, K\} \\ R_{Chegada} & e \in Cheg^{Veículo}, a = -1 \\ -P_{Deixa} & e \in Deixa_k^{Veículo}, a = -1 \\ R_{Ter} & e \in Fim_k^{Req}, a = -1 \\ 0 & \text{Caso Contrário} \end{cases} \quad (8)$$

Na equação 8 a variável $R_{Chegada}$ representa a recompensa do sistema pela chegada de um novo veículo que compartilhará os seus recursos, sendo descrita como $R + \sum_k^K b_k$. A variável R_{Req} representa a recompensa do sistema pelo aceite da requisição do serviço, sendo descrita como $R - b_k$, onde o R é o ganho do sistema por uso dos recursos da *cloud*. Embora haja uma recompensa quando há alocação de recurso na *cloud*, o serviço aceito requerirá b_k recursos da *cloud* veicular, diminuindo os recursos disponíveis. A variável R_{Ter} representa a recompensa do sistema pelo término de execução de um serviço, sendo descrita como $R + b_k$.

As variáveis P_{Deixa} e P_{Req} representam a penalidade do sistema devido à saída de algum veículo da *cloud* ou devido à um bloqueio de alguma requisição de serviço feita para a *cloud*. A variável P_{Deixa} representa a penalidade devido à saída do veículo que está compartilhando seus recursos. Nesse evento a *cloud* perde um conjunto de recursos que poderia ser utilizado pelos seus usuários ($P_{Deixa} = P - b_k$). A variável P representa a penalidade do sistema. A variável P_{Req} representa a penalidade do sistema devido ao bloqueio da requisição de algum serviço.

Embora o recurso do veículo que requisita o serviço possa estar disponível para a *cloud*, o sistema é penalizado devido à sobrecarga de mensagens causada pela requisição. Assim, P_{Req} é definido como $P - \sum_{k=1}^K b_k$.

O custo esperado do sistema $g(s, a)$ é definido como:

$$g(s, a) = \sum_{k=1}^K ((w_u * b_k * n_k^{Req} * \gamma(s, a)) + (w_a (\frac{n^{Res}}{b_k}))) \quad (9)$$

onde w_u e w_a são fatores de ponderação que indicam a importância relativa entre a utilização de recursos e a disponibilidade de recursos, sendo $w_u + w_a = 1$.

4.5. Solução do SMDP

O MAVERIC é baseado no modelo de recompensa médio [Puterman 1994] que maximizará a recompensa média esperada do sistema pelo algoritmo iterativo. A recompensa média da política π , sendo π uma política factível, é definida como:

$$g^\pi = \lim_{N \rightarrow \infty} \frac{E^\pi \left\{ \sum_{n=1}^N r_n(s, a) \right\}}{E^\pi \left\{ \sum_{n=1}^N \tau_n(s, a) \right\}} \quad (10)$$

$$g^* = g^{\pi^*} = \sup_{\pi \in \Pi} g^\pi \quad (11)$$

onde Π é o conjunto de π . Assumindo $v(s)$ como sendo uma função potencial do estado s , a política ótima π que satisfaz a equação 11 pode ser obtida resolvendo a equação de Bellman recursivamente, i.e.,

$$v(s) = \max_{a \in A_s} \left\{ r(s, a) - g\tau(s, a) + \sum_{s' \in S} p(s'|s, a)v(s') \right\}, \forall s \in S \quad (12)$$

Depois disso, precisamos transformar este modelo em um modelo de tempo discreto equivalente. Para isso, usamos a transformação de uniformização que é dada por:

$$\tilde{r}(s, a) = r(s, a) \frac{\sigma(s, a)}{y}, \quad (13)$$

$$\tilde{g} = \frac{g}{y}, \quad (14)$$

$$\tilde{p}(s'|s, a) = \begin{cases} 1 - \frac{[1-p(s|s,a)]\sigma(s,a)}{y} & s' = s \\ \frac{p(s'|s,a)\sigma(s,a)}{y} & s' \neq s \end{cases} \quad (15)$$

Por fim, o Bellman segue como:

$$\tilde{v}(s) = \max_{a \in A_s} \left\{ \tilde{r}(s, a) - \tilde{g} + \sum_{s' \in S} \tilde{p}(s'|s, a)\tilde{v}(s') \right\}, \forall s \in S \quad (16)$$

Depois que o modelo proposto é transformado em um SMDP com espaço de estados e ação finitos, a iteração de valores pode ser usada para encontrar a política ótima. O fluxo do algoritmo iterativo pode ser visto em detalhes no Algoritmo 1.

Algorithm 1 Algoritmo Iterativo

1: $\tilde{v}(s) = 0$ para cada estado s . Especificamos $\varepsilon > 0$ e $n = 0$.

2: **for** cada $s \in S$, calcule $\tilde{v}^{n+1}(s)$ por **do**

3:

$$\tilde{v}^{(n+1)}(s) = \max_{a \in A_s} \left\{ \tilde{r}(s, a) + \sum_{s' \in S} \tilde{p}(s'|s, a) \tilde{v}^n(s') \right\} \quad (17)$$

4: **end for**

5: **if** ($sp(\tilde{v}^{n+1} - \tilde{v}^n) < \varepsilon$) **then**

6: **for** cada $s \in S$, escolha **do**

7:

$$d_\varepsilon(s) \in \arg \max_{a \in A_s} \left\{ \tilde{r}(s, a) + \sum_{s' \in S} \tilde{p}(s'|s, a) \tilde{v}(s') \right\} \quad (18)$$

8: **end for**

9: **else**

10: $n = n + 1$ e volte para a linha 2.

11: **end if**

5. Análise de Desempenho

O MAVERIC foi implementado em Matlab [Cros 2015] com o intuito de avaliar o desempenho do método proposto, sendo que para isso assumiu-se que o sistema será executado em uma cidade inteligente, na qual possui uma quantidade relevante de veículos trafegando nas vias e, eventualmente, este fluxo de veículos pode causar um engarrafamento em algum período do dia.

O poder de transmissão desses veículos é de aproximadamente 0.98 mW. Esse poder de transmissão permite uma área de comunicação de aproximadamente 200 metros. Cada veículo tem o tamanho de 4 metros. Os veículos se comunicarão dentro do *cluster*. Esses *clusters* (*clouds*) têm uma área de cobertura de 200 metros (o tamanho da área de comunicação do *cluster*). A quantidade de veículos de cada *cluster* é dada por $\text{área_de_cobertura}/\text{tamanho_do_veículo}$. Portanto, a *cloud* terá um total de 50 veículos que poderão compartilhar seus recursos.

Com base na quantidade de recursos disponíveis na *cloud* veicular, ela pode oferecer um conjunto de serviços que serão utilizados pelos seus usuários, tais como, processamento, rede, armazenamento, entre outros serviços. Neste trabalho, consideramos que cada *cloud* oferecerá dois tipos de serviços e que cada serviço necessitará de uma quantidade de recursos alocados para poder ser executado. Como exemplo, podemos considerar que uma *cloud* oferecerá armazenamento e processamento, e que esses serviços precisarão alocar um conjunto de recursos que variam entre 2, 3, 4 e 5 recursos. Cada requisição de serviço pode variar entre 5, 10, 15, 20 e 25 requisições por segundo. A Tabela 2 descreve os principais parâmetros utilizados em nossa análise.

O objetivo dessa avaliação é analisar o comportamento do MAVERIC para verificar quais ações que o sistema tomará para maximizar a disponibilidade dos serviços. Para isso, também variamos o tempo de execução de cada serviço.

Parâmetros	Valores
$\lambda_{chegada}$	1, 5, 10, 15, 20
$Deixar_{cloud}$	5, 10, 15, 20, 25
λ_K	2, 4, 6, 8, 12
μ_k	0.1, 0.2, 0.4, 0.6, 0.8
w_u	0.25
w_a	0.75
K	2
b_k	2, 3, 4, 5
R	10
P	-10

5.1. Métricas de Avaliação

Neste Trabalho, utilizamos duas métricas para avaliar o comportamento de MAVeRIC:

- A **Recompensa Média do Sistema** em diferentes taxas que chegam e saem de veículos e solicitações de serviço;
- A **Porcentagem de Bloqueio** de serviço da classe k .

5.2. Resultados

A Figura 2(a) mostra a média de recompensa do sistema quando variamos o λ_1 e mantemos os valores de $\lambda_{chegada} = 10$, $Deixar_{cloud} = 15$, $\lambda_2 = 2$, μ_1 e $\mu_2 = 0.1$. Podemos observar que ao variarmos a chegada de novas requisições da classe, a recompensa do sistema tende a aumentar em virtude da função de recompensa, que apesar de o sistema estar alocando b_k recursos, ele está atendendo a mais solicitações. Portanto, o sistema está conseguindo disponibilizar mais recursos para atender a demanda solicitada. Esse comportamento também é observado quando variamos o λ_2 , como pode ser visto na Figura 2(b).

A Figura 2(b) mostra a média de recompensa do sistema quando variamos o λ_2 , e mantemos os valores de $\lambda_{chegada} = 10$, $Deixar_{cloud} = 15$, $\lambda_2 = 2$, μ_1 e $\mu_2 = 0.1$. Observando os pontos entre 2 e 5 requisições por segundo podemos observar que os cenários com $b_1 = 3$ e $b_2 = 4$, e $b_1 = 4$ e $b_2 = 3$ possuem o mesmo comportamento devido ao baixo aumento da frequência das requisições no sistema. Entretanto, acima de 5 requisições por segundo o valor obtido na probabilidade de transição sofre um impacto significativo devido à diferença entre a quantidade de recursos alocados, e também devido ao aumento da frequência da requisição da classe 2.

A Figura 2(c) apresenta a média de recompensa do sistema quando variamos a frequência de entrada de veículos na *cloud* mantendo os parâmetros a seguir: $Deixar_{cloud} = 15$, λ_1 e $\lambda_2 = 2$, μ_1 e $\mu_2 = 0.1$. Podemos observar que com a baixa frequência de entrada de veículos na *cloud*, a recompensa do sistema cai, pois os recursos disponíveis estarão alocados devido à alta demanda de serviço e o aparecimento de novos recursos vai depender exclusivamente da entrada de novos veículos. Se observarmos o cenário com $b_1 = 2$ e $b_2 = 3$ a recompensa foi a 0, pois com baixa quantidade de recursos

para atender à duas classes de serviço, os recursos foram alocados rapidamente. Desta forma, o sistema encontra-se sem recursos disponíveis na *cloud*.

A Figura 2(d) apresenta a média de recompensa do sistema quando variamos a probabilidade de saída do veículo da *cloud* e mantendo os valores de $\lambda_{chegada} = 10$, λ_1 e $\lambda_2 = 2$, μ_1 e $\mu_2 = 0.1$. Com o aumento da probabilidade dos veículos deixarem a *cloud*, a recompensa da *cloud* diminui, pois o veículo, ao sair, retira seus recursos da *cloud*. Além disso, caso o veículo tenha solicitado algum recurso, a rede sofrerá uma sobrecarga para que esse serviço seja deslocado para a outra *cloud*. Podemos observar que para serviços que precisam de muitos recursos alocados para poderem ser executados, a queda de recompensa é mais lenta, pois apenas poucos recursos serão realocados para outros veículos. Já em serviços que precisam de poucos recursos a queda é mais rápida, pois o serviço poderá ser requisitado novamente em uma nova *cloud*, ou seja, todo o serviço precisará ser migrado.

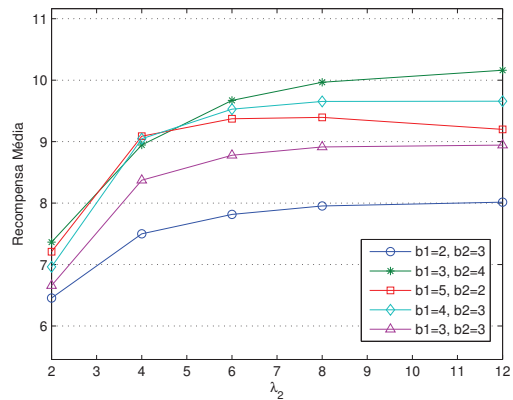
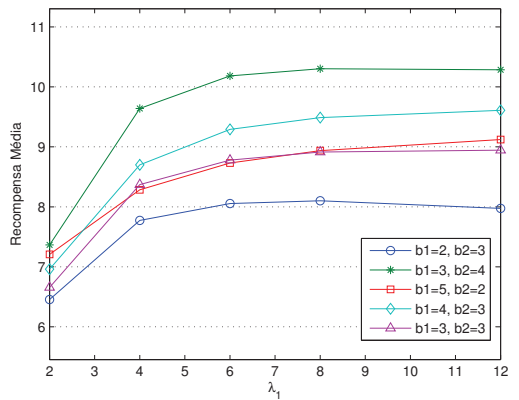
A Figura 2(e) e a Figura 2(f) apresenta a média de recompensa do sistema quando variamos o tempo de execução do serviço da classe 1 (μ_1) e da classe 2 (μ_2) respectivamente, mantendo os valores de $\lambda_{chegada} = 10$, $Deixar_{cloud} = 5$, λ_1 e $\lambda_1 = 2$, $\mu_2 = 0.1$ quando variamos μ_1 e $\mu_1 = 0.1$ quando variamos μ_2 . Ambos os gráficos, possuem comportamento similar, pois o que realmente impacta na probabilidade de transição entre um estado e outro é a quantidade de recursos que uma determinada classe de aplicação necessita para poder executar. Assim, a variação do tempo de execução de cada classe de serviço impactará no tempo esperado para a próxima tomada de decisão, mas não tendo um grande impacto na recompensa do sistema. Portanto, como podemos observar no gráfico somente uma queda em sua recompensa.

O tempo de execução dos serviços não teve um impacto significativo no momento da tomada de decisão do sistemas, como podemos ver na Figura 3. Essa figura mostra a probabilidade de ação de bloqueio pelo sistema quando variamos o tempo de execução dos serviços solicitados. Para isso, mantemos os valores de $\lambda_{chegada} = 10$, $Deixar_{cloud} = 5$, λ_1 e $\lambda_1 = 2$, e $\mu_2 = 0.1$. A baixa probabilidade de ação de bloqueio é decorrente à alta taxa de chegada de veículos na *cloud*, bem como a baixa probabilidade desses veículos deixarem a *cloud* veicular. Portanto, o MAVeRIC demonstrou uma boa eficiência em sua tomada de decisão em alocar ou não os recursos disponíveis na *cloud*.

O resultado numérico mostra que o MAVeRIC é estável independentemente da taxa de solicitação de serviço ou da taxa de chegada e saída do veículo. Além disso, o MAVeRIC tem uma taxa de bloqueio com cerca de 2%, mostrando que o MAVeRIC maximiza a disponibilidade de recursos e sua utilização.

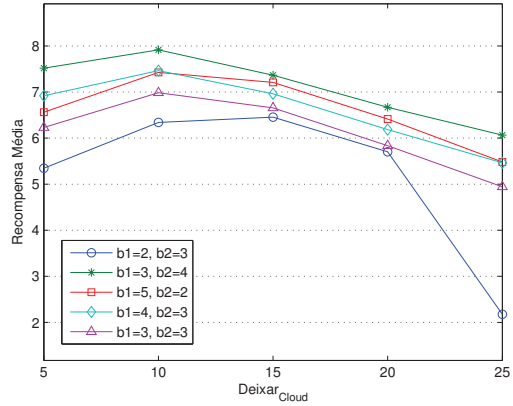
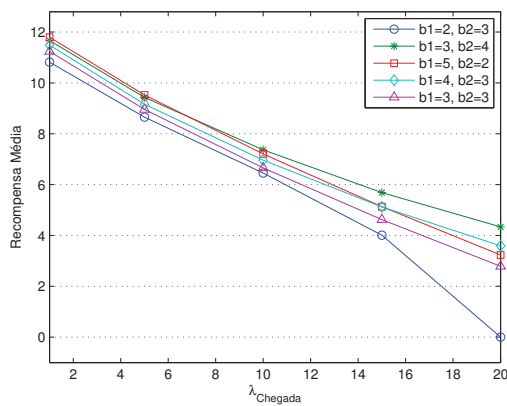
6. Conclusão

Neste trabalho, propusemos o MAVeRIC, que aborda o problema de gerenciamento e alocação de recursos em uma *cloud* veicular, considerando que os veículos possuem recursos e que esses possam ser compartilhados, aumentando assim o número de serviços oferecidos pela *cloud*. O MAVeRIC foi formulado como Semi-Markov Decision Process. A otimização da tomada de decisão é obtida por meio do algoritmo iterativo, a fim de maximizar a recompensa média esperada do sistema e melhorar a disponibilização dos recursos. Os resultados numéricos mostram que o desempenho de recompensa esperado é significativo, estável, com menor taxa de bloqueio e boa decisão com a relação de alo-



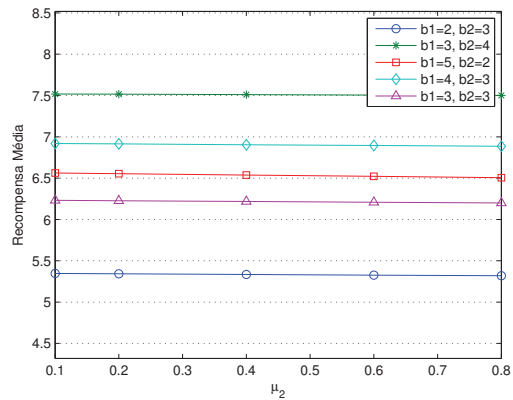
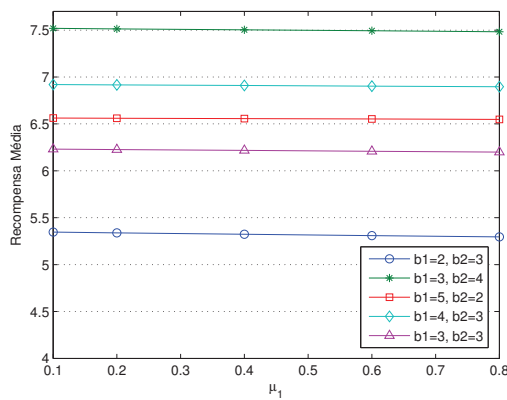
(a) Média de recompensa do sistema quando variamos o λ_1

(b) Média de recompensa do sistema quando variamos o λ_2



(c) Média de recompensa do sistema quando variamos a frequência de entrada de veículos na cloud

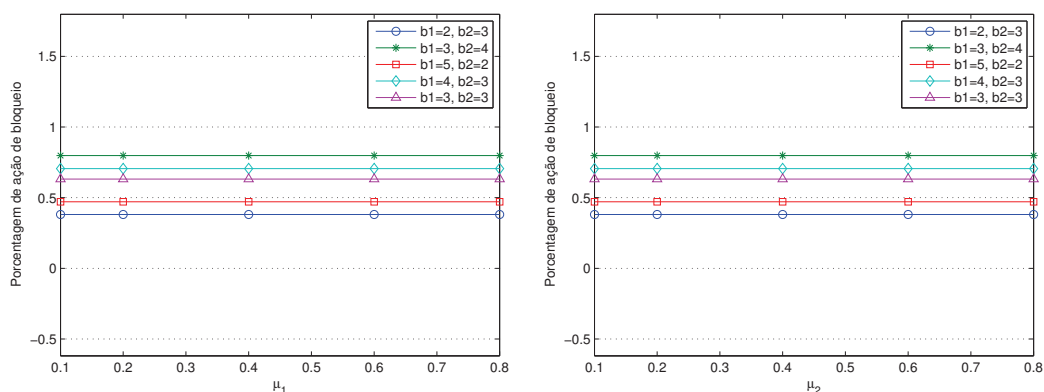
(d) Média de recompensa do sistema quando variamos a probabilidade de saída do veículo da cloud



(e) Média de recompensa do sistema quando variamos o tempo de execução do serviço da classe 1 (μ_1)

(f) Média de recompensa do sistema quando variamos o tempo de execução do serviço da classe 2 (μ_2)

Figura 2. Desempenho do MAVeRIC considerando a Recompensa Média do Sistema.



(a) Probabilidade de ação de bloqueio pelo sistema quando variamos o tempo de execução do serviço da classe 1 (b) Probabilidade de ação de bloqueio pelo sistema quando variamos o tempo de execução do serviço da classe 2

Figura 3. Desempenho do MAVeIC considerando a probabilidade da ação de bloqueio do sistema.

cação de recursos.

7. Agradecimento

Os autores agradecem a Fundação de Amparo à Pesquisa do Estado de São Paulo - FAPESP (processo número 2015/11536-4 e o processo número 2015/18898-9) pelo apoio financeiro.

Referências

- Arkian, H. R., Atani, R. E., Diyanat, A., and Pourkhalili, A. (2015). A cluster-based vehicular cloud architecture with learning-based resource management. *The Journal of Supercomputing*, 71(4):1401–1426.
- Ashokkumar, K., Sam, B., Arshadprabhu, R., and Britto (2015). Cloud based intelligent transport system. *Procedia Computer Science*, 50:58 – 63.
- Cheng, X., Yang, L., and Shen, X. (2015). D2d for intelligent transportation systems: A feasibility study. *IEEE Transactions on Intelligent Transportation Systems*, 16(4):1784–1793.
- Chirayil, G. S. and Thomas, A. (2016). A study on cost effectiveness and security of {VANET} technologies for future enhancement. *Procedia Technology*, 25:356 – 363. 1st Global Colloquium on Recent Advancements and Effectual Researches in Engineering, Science and Technology - {RAEREST} 2016 on April 22nd, 23rd April 2016.
- Cros, M. J. (2015). Markov decision processes (mdp) toolbox. last visited in September, 2016,.
- Dai, P., Liu, K., Sha, E., Zhuge, Q., Lee, V., and Son, S. H. (2015). Vehicle assisted data update for temporal information service in vehicular networks. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, pages 2545–2550.

- Farooq, M. U., Pasha, M., and Khan, K. U. R. (2016). Contextual communication patterns for vehicular cloud computing: Towards realizing internet of things. In *2016 10th International Conference on Intelligent Systems and Control (ISCO)*, pages 1–6.
- Ghazal, A., Wang, C. X., Ai, B., Yuan, D., and Haas, H. (2015). A nonstationary wide-band mimo channel model for high-mobility intelligent transportation systems. *IEEE Transactions on Intelligent Transportation Systems*, 16(2):885–897.
- Ghazizadeh, P., Florin, R., Zadeh, A. G., and Olariu, S. (2016). Reasoning about mean time to failure in vehicular clouds. *IEEE Transactions on Intelligent Transportation Systems*, 17(3):751–761.
- Khan, N. A., Brujic-Okretic, V., and Khaddaj, S. (2016). Intelligent integration framework for smart transport system. In *2016 12th International Conference on Intelligent Environments (IE)*, pages 76–79.
- Liu, K., Ng, J. K. Y., Lee, V. C. S., Son, S. H., and Stojmenovic, I. (2016). Cooperative data scheduling in hybrid vehicular ad hoc networks: Vanet as a software defined network. *IEEE/ACM Transactions on Networking*, 24(3):1759–1773.
- Meneguetto, R., Boukerche, A., and De Grande, R. (2016). SMART: an efficient resource search and management scheme for vehicular cloud-connected system. In *2016 IEEE Global Communications Conference: Mobile and Wireless Networks (Globecom2016 MWN)*, Washington, USA.
- Meneguetto, R. I. (2016). A vehicular cloud-based framework for the intelligent transport management of big cities. *International Journal of Distributed Sensor Networks*, 12(5).
- Meng, H., Zheng, K., Chatzimisios, P., Zhao, H., and Ma, L. (2015). A utility-based resource allocation scheme in cloud-assisted vehicular network architecture. In *2015 IEEE International Conference on Communication Workshop (ICCW)*, pages 1833–1838.
- Miao, Z., Li, C., Zhu, L., Han, X., Wang, M., Cai, X., Liu, Z., and Xiong, L. (2016). On resource management in vehicular ad hoc networks: A fuzzy optimization scheme. In *2016 IEEE 83rd Vehicular Technology Conference (VTC Spring)*, pages 1–5.
- Puterman, M. L. (1994). *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley and Sons, Inc., New York, NY, USA, 1st edition.
- Pyun, S. Y., Lee, W., and Cho, D. H. (2016). Resource allocation for vehicle-to-infrastructure communication using directional transmission. *IEEE Transactions on Intelligent Transportation Systems*, 17(4):1183–1188.
- Sladkowski, A. and Pamula, W. (2015). *Intelligent Transportation Systems - Problems and Perspectives*. Springer Publishing Company, Incorporated.
- Yu, R., Huang, X., Kang, J., Ding, J., Maharjan, S., Gjessing, S., and Zhang, Y. (2015). Cooperative resource management in cloud-enabled vehicular networks. *IEEE Transactions on Industrial Electronics*, 62(12):7938–7951.
- Zheng, K., Meng, H., Chatzimisios, P., Lei, L., and Shen, X. (2015). An smdp-based resource allocation in vehicular cloud computing systems. *IEEE Transactions on Industrial Electronics*, 62(12):7920–7928.