Aprendizado por Reforço para Escalonamento de Recursos em Sistema sem Fio Multiportadora com Ondas Milimétricas Utilizando Modelo Markoviano

Daniel Porto Queiroz Carneiro[1], **Alisson Assis Cardoso**[1],
Cláudio Gabriel Lemos de Almeida [1], **Flávio Henrique Teles Vieira**[1]

[1]Escola de Engenharia Elétrica, Mecânica e de Computação
Universidade Federal de Goiás (UFG)

Abstract. *In this paper, a resource allocation algorithm based on reinforcement learning is presented for a multicarrier communication system considering multiple users, fading and multipath effects in a transmission assuming millimeter waves. To this end, it is proposed that the communication system can be described by a Markovian model represented by the states of the queue in the buffers and states of the channels. For the resource allocation algorithm in this work, we introduce a different reward function used in the reinforcement learning Q-learning algorithm. The results obtained in the simulations show that the application of the proposed resource scheduling algorithm generally provides an improvement in the performance parameters of the considered communication system, such as an increase in its throughput and decrease of lost packets. Comparisons with other algorithms presented in the literature are carried out, also showing that the use of the proposed reward function and Markovian model makes user scheduling and resource sharing more efficient.*

Resumo. *Neste artigo, apresenta-se um algoritmo de alocação de recursos baseado em aprendizado por reforço para um sistema de comunicação multiportadora considerando múltiplos usuários e efeitos de desvanecimento e multipercurso em uma transmissão assumindo ondas milimétricas. Para tal, propõe-se que o sistema de comunicação possa ser descrito por um modelo Markoviano representado pelos estados da fila nos buffers e estados dos canais. Para o algoritmo de alocação de recursos deste trabalho, introduzimos uma função de recompensa a ser utilizada no algoritmo de aprendizado por reforço Q-learning. Os resultados obtidos nas simulações mostram que a aplicação do algoritmo proposto de escalonamento de recursos provê de forma geral, melhoria nos parâmetros de desempenho do sistema de comunicação considerado, como por exemplo, aumento de vazão e diminuição de perda de pacotes. Comparações com outros algoritmos apresentados na literatura são realizadas, mostrando também que o uso da função de recompensa e o modelo Markoviano propostos torna o escalonamento de usuários e o compartilhamento de recursos mais eficientes.*

1. Introdução

Um dos desafios em sistemas de comunicação é compartilhar recursos de forma eficiente sendo estes limitados. Além da faixa de frequências disponíveis de transmissão ser finita, a potência utilizada é um fator limitante especialmente em dispositivos com baterias. Com a demanda cada vez mais alta de qualidade, alta taxa de transmissão e com o crescimento de usuários e dispositivos, uma estratégia adequada de alocação de recurso se mostra imperativa.

Os sistemas de comunicação são complexos e podem priorizar um indicador de desempenho específico em detrimento a outros, por exemplo, aumentar a vazão sem se preocupar com gasto de potência, atender equipamentos mais próximos e postergar atendimento de equipamentos mais distantes. Os autores [Zhu et al. 2018] propõem um algoritmo de aprendizado por reforço aplicado a um sistema IoT (*Internet of Things*) multiusuários. O agente único de controle atende a um usuário de cada vez, multiplexando o atendimento no tempo. Apesar de considerar velocidade relativa entre transmissor e receptor, não são abordadas as distâncias entre eles ou suas velocidades absolutas.

No artigo [Ford et al. 2017], aborda-se o desempenho de um sistema LTE OFDM (*Long Term Evolution Orthogonal Frequency Division Multiplexing*) onde são consideradas mais informações para os ganhos dos canais como situação de inoperância (*outage*). Além disso, são comparadas estatísticas de dados reais de um sistema de comunicação em ambiente urbano com resultados de um Modelo Markoviano finito de canal aplicados ao sistema LTE-OFDM considerado. Em sistemas OFDM, o atendimento dos usuários é feito simultaneamente em diferentes frequências. Com isso, o desempenho do sistema pode apresentar maiores valores de vazão, menores valores de tempo de espera dos pacotes na fila no *buffer* e menores valores de perda de pacotes do que sistemas que não consideram múltiplas subportadoras [Patteti et al. 2016].

Neste artigo, considera-se um sistema OFDM de comunicação com um agente inteligente de controle baseado em aprendizado por reforço. Para este sistema de comunicação é imposta uma restrição de BER (*Bit Error Rate* - Taxa de erro de bit) para se obter a potência mínima necessária em um ambiente cuja propagação é realizada por ondas milimétricas. Assume-se um modelo TDL (*Tapped Delay Line*) para a modelagem deste canal de comunicação com característica estocástica para os ganhos [Zhu et al. 2018]. Neste artigo, utiliza-se o algoritmo *Q-learning* com iteração de política e modelo Markoviano para os estados do sistema de comunicação. O modelo de canal utilizado segue a configuração apresentada pelos autores em [Hong Shen Wang and Moayeri 1995] e [3GPP 2018]. Os diferentes valores dos ganhos dos canais levam em conta perdas por múltiplo percurso e falta de linha de visada. Quanto à avaliação de QoS (*Quality of Service*), assim como em [Zhu et al. 2018] avalia-se a vazão de pacotes, a perda de pacotes, ocupação de pacotes na fila do *buffer* e eficiência energética, entretanto para um sistema multiportadora e considerando uma modelagem de canal mais apropriada para a tecnologia 5G.

Como principais contribuições deste artigo, pode-se citar:

1. Apresentação de algoritmo de aprendizado por reforço adaptado para otimizar parâmetros de qualidade de serviço em um sistema de comunicação OFDM;
2. Proposta de função de utilidade para o algoritmo de aprendizagem por reforço;

3. Avaliação da Utilização de *Q-Learning off-policy* baseado em modelo Markoviano do sistema considerando os estados do *buffer* e do canal.

2. Modelo do Sistema OFDM

O modelo de sistema OFDM considerado neste artigo consiste em uma estação rádio base (agente) que deve tomar decisões a cada *frame* (intervalo de tempo, utilizado igual a 2 ms) sobre quando e como atender K equipamentos de usuários. Para isso, o sistema de comunicação utiliza M canais e J modos de transmissão através de um sistema de subportadoras (OFDM) apresentando um *buffer* de tamanho L para cada usuário.

O agente é treinado utilizando aprendizado por reforço (*Q-learning*) com base nos estados possíveis do sistema e uma função de recompensa. Para descrever os estados do sistema de comunicação, adota-se um modelo Markoviano, isto é, a mudança de estado exige o conhecimento do estado atual do sistema, da ação escolhida e do ambiente (característica estocástica). Assim, são descritos a seguir como são estimados os estados do *buffer*, do canal e como a alocação de potência para os usuários é efetuada.

2.1. Estados do Buffer

O *buffer* possui tamanho L, ou seja tem capacidade de armazenar no máximo L pacotes para cada usuário. Os estados do sistema são obtidos levando em conta os $L + 1$ estados possíveis, com inclusão do zero, para cada um dos K usuários. Dessa forma, para K usuários cujos dispositivos apresentam *buffer* de tamanho L, tem-se $(L + 1)^K$ estados de *buffer* possíveis no sistema. Há mudanças no estado do *buffer* com a chegada ou saída de dados, o que pode se modificar a cada *frame*.

2.2. Estados do Canal

Após deixar o transmissor, o sinal se propaga no ambiente e irá chegar ao receptor com características diferentes de quando partiu. A seguir são tratados dois efeitos na amplitude e potência do sinal considerando que não há um caminho direto (visada direta) entre o transmissor e receptor.

2.2.1. Múltiplos Percursos

Por não ter um caminho direto (linha de visada) para percorrer, o sinal que chega ao receptor passa por reflexões diversas. Assim, existem múltiplos percursos do sinal. Considerando um ambiente de ruído AWGN (*Additive White Gaussian Noise*), a amplitude do sinal (v) pode ser modelada por uma variável aleatória caracterizada por uma distribuição de Rayleigh [Matz and Hlawatsch 2011]:

$$p_v(v) = \frac{v}{\sigma^2} e^{\frac{-v^2}{2\sigma^2}} \tag{1}$$

onde $p_v(v)$ é a densidade de probabilidade da amplitude do sinal v.

A distribuição de probabilidade da potência $v^2 = |h|^2.P$ no equipamento é, portanto, exponencial [Proakis and Salehi 2008]. Assumindo que a P (Potência na estação base) e WN_0 sejam constantes no *frame* (2 ms), a SNR é dada por:

$$SNR = |h|^2 \frac{P}{WN_0} \tag{2}$$

onde h representa o coeficiente de ganho do canal e WN_0 a densidade de potência do ruído.

Seja ρ_m o ganho médio de potência do canal, então a probabilidade do ganho médio do canal ser ρ é dada por:

$$p_\rho(\rho) = \frac{1}{\rho_m} e^{\frac{-\rho}{\rho_m}} \tag{3}$$

Neste artigo, foi considerado o modelo TDL (*Tapped Delay Line*) para caracterizar um canal com características variantes no tempo, conforme descrito em [3GPP 2018].

2.2.2. Desvanecimento

Além do efeito dos múltiplos percursos no ganho do canal, a distância entre transmissor e receptor também é um fator limitante no seu valor. O modelo de desvanecimento utilizado segue a equação de *Path Loss*, PL como em [3GPP 2018]:

$$PL = 32.4 + 20log(fc) + 30log(D) \tag{4}$$

onde fc (GHz) é a frequência da portadora e D (m) a distância entre transmissor e o receptor. O ganho final do canal considerando os dois efeitos pode ser representado por,

$$|h|^2 = \frac{\rho}{10^{PL/10}} \tag{5}$$

onde ρ é o ganho do canal pelo efeito de múltiplo percurso, sendo uma variável aleatória.

2.3. Potência

A potência $P(c, j)$ do canal c é calculada explicitando o termo $P(c, j)$ da equação de BER (taxa de erro de bit) máxima conforme mostram as equações (6) e (7) que dependem do modo j e da potência do ruído WN_0. Neste trabalho, se considera um valor de potência alocada ao dispositivo do usuário de tal modo a garantir uma BER igual ou maior do que 0.001. Para atendimento da demanda de tráfego dos usuários, utiliza-se 3 modos diferentes de transmissão BPSK, 4QAM e 8QAM proporcionando diferentes taxas de geração de pacotes.

- $BPSK, j = 1$

$$P(c, j) \geq \frac{inverfc(BER(c, j).2)^2}{\rho/WN_0} \tag{6}$$

- $2^j - QAM, j > 1$

$$P(c, j) \geq \frac{(2^j - 1)ln(5BER(c, j))}{-1.6\rho/WN_0} \tag{7}$$

3. Aprendizado por Reforço Markoviano para Escalonamento de Recursos

O sistema descrito na seção anterior, pode ser modelado como uma cadeia de Markov, considerando que o estado seguinte dependa somente do estado atual e da ação escolhida pelo agente [Zhu et al. 2018].

O aprendizado por reforço é uma técnica que consiste em um agente tomando decisões em diversos estados de um ambiente e recebendo recompensas ou punições pelas suas ações [Sutton and Barto 2018]. Após uma série de testes de tentativa-erro, o agente busca aprender a melhor política, ou seja, a melhor sequência de ações a serem tomadas naquele ambiente de forma a obter valores de recompensas maiores.

Nesse artigo, o algoritmo de aprendizado por reforço *Q-learning* é utilizado, no qual é necessário obter as probabilidades de transição de estados e as recompensas de cada ação possível.

3.1. Ações OFDM

Cada ação é composta pela escolha do usuário e do modo de transmissão para cada um dos M canais. Estas ações se referem ao cenário onde além de atendimento individual, permite-se atendimento simultâneo através de ações OFDM.

A multiplexação de frequência ortogonal permite transmitir simultaneamente em M canais utilizando subportadoras. Assim, cada ação é composta de M dos K usuários e M dos $(J+1)$ modos de transmissão. A quantidade de ações possíveis será a permutação M dos $(J+1)$ multiplicada pela permutação M dos K usuários:

$$Na = \frac{K!}{(K-M)!}\frac{(J+1)!}{(J+1-M)!} \tag{8}$$

3.2. Transição de Estados

A transição de estados por ação ocorre segundo um processo de decisão de Markov. Considera-se que o novo estado só depende do estado anterior da ação utilizada e do ambiente que define a chegada de dados e ganhos do canal para o *frame* corrente. Os estados possíveis combinam os diferentes estados de *buffer* $S_b = (L+1)^K$ e diferentes estados de canais $S_c = C^M$ de forma a obter $S_b \cdot S_c$ estados possíveis. Por serem independentes, a probabilidade de transição é o produto das probabilidades de transição do *buffer* e do canal.

3.2.1. Estados do *buffer*

Uma das informações de entrada para o modelo Markoviano, a taxa média de pacotes, está relacionada com a quantidade b de pacotes gerados. Assume-se que a geração de pacotes obedeça a uma distribuição de Poisson com taxa média de chegada λ para cada um dos K usuários. Ou seja, tem-se a seguinte equação para a probabilidade de ocorrência de b pacotes:

$$Prob(b, \lambda) = \frac{e^{-\lambda}\lambda^b}{b!} \tag{9}$$

Durante o *frame* i, o *buffer* do usuário k possui l pacotes. Se chegam b pacotes a este *buffer* e t_a pacotes são transmitidos com a ação a, o novo estado do *buffer* pode ser dado por:

$$l_{i+1} = min(l_i + b - t_a, L) \tag{10}$$

Assim, pode-se reescrever (11) como:

$$pb_k(l_i, l_{i+1}|a) = \frac{e^{-\lambda}\lambda^b}{b!} \tag{11}$$

Como não há dependência entre os usuários, tem-se que:

$$pb(b, b^{'}) = \prod_{k=1}^{K} pb_k(b, b^{'}|a) \tag{12}$$

sendo b e $b^{'}$ estados do conjunto combinado de $(L+1)^K$ estados. Portanto, a matriz p_b de transição de estados do *buffer* depende da ação escolhida (que define t_a) e, portanto, é uma matriz S_b x S_b x Na.

3.2.2. Estados do canal

O ambiente simulado possui C estados possíveis para cada um dos M canais $c_0, ...c_{C-1}$ de acordo com o ganho $|h|^2$ do canal e $C-1$ limiares. $\rho = \rho_1, \rho_2, ...\rho_{C-1}$, com $\rho_0 = 0$ e $\rho_C = inf$. A probabilidade do canal estar no estado c_n é dada por:

$$pc(c_n) = \int_{\rho_n}^{\rho_{n+1}} pdf(\rho)d\rho \tag{13}$$

Da equação (3), obtém-se:

$$pc(c_n) = e^{\frac{-\rho_n}{\rho_m}} - e^{\frac{-\rho_{n+1}}{\rho_m}} \tag{14}$$

A transição de estados dos canais se dá apenas entre estados vizinhos da cadeia de Markov de nascimento e morte, de forma independente para cada canal:

$$pc_m(c_n, c_{n+1}) = \frac{N(c_{n+1})Tf}{pc(c_n)} \tag{15}$$

$$pc_m(c_n, c_{n-1}) = \frac{N(c_n)Tf}{pc(c_n)} \tag{16}$$

onde Tf é a duração do *frame* em segundos e $N(c_n)$ é o numero de vezes que o limiar c_n é cruzado por segundo, conforme [Rappaport et al. 1996]:

$$N(c_n) = \sqrt{\frac{2\pi\rho_{c_n}}{\rho_m}} f_D e^{-\rho_n/\rho_m} \tag{17}$$

em que f_D é o máximo efeito Doppler.

Dado que são M canais, cada um com C estados possíveis e pode-se utilizar mais de um canal ao mesmo tempo com o atendimento OFDM, tem-se C^M estados possíveis e independentes. Logo,

$$pc(c, c^{'}) = \prod_{m=1}^{M} pc_m(c(m), c(m)^{'}) \tag{18}$$

sendo c e $c^{'}$, estados do conjunto combinado de C^M estados, e $c(m)$ e $c(m)^{'}$ o estado individual de cada canal, do conjunto de C estados. Como assume-se que a transição de estados dos canais siga um modelo Markoviano, $pc_m(c(m), c(m)^{'})$ é zero para estados não vizinhos.

Assim, a probabilidade total de transição de estados para ação a é dada por:

$$ps(S, S^{'}|a) = \prod_{k=1}^{K} pb_k(b, b^{'}|a) \prod_{m=1}^{M} pc_m(c(m), c(m)^{'}) \tag{19}$$

3.3. Função Utilidade

A função utilidade é responsável por agrupar as variáveis do sistema e descrever a relação entre elas de forma que permita a solução do processo de aprendizado por reforço convergir para regiões com desempenho desejado. Neste artigo, os parâmetros utilizados na função utilidade são o fluxo de dados $B_k(s, a)$ em pacotes do usuário k e o custo $C_k(s, a)$ do usuário k composto pelo consumo de potência e a pressão total dos usuários no *buffer* ou de pacotes perdidos. A função utilidade (ou de recompensa) é dada por:

$$R(s, a) = \sum_{k=1}^{K} \frac{B_k(s, a)}{C_k(s, a)} \tag{20}$$

$$B_k(s, a) = min(l_k + b, V \cdot j) \tag{21}$$

onde l_k é quantidade de pacotes no *buffer*, b é o número de pacotes que chegaram, V é a taxa de código (*code rate*) e $j = 0...J$ o modo de transmissão.

[Zhu et al. 2018] considera o termo $B_k(s, a) = V.j$. Essa consideração supõe que a transmissão de pacotes é sempre a máxima possível (capacidade do sistema) para a função de recompensa. Para a proposta, o cálculo da matriz R considerada uma taxa média de chegada λ e a quantidade de pacotes transmitidos é ponderada pela probabilidade de chegar b pacotes $B_k(s, a) = E[min(l + b_\lambda, V.j)]$. Na simulação do atendimento foi respeitada a distribuição de poisson para taxa de chegada, ou seja, a cada intervalo de tempo toma-se uma amostra a distribuição para cada usuário.

Neste trabalho, propõe-se adotar na função utilidade proposta, a soma das razões $\sum(B/C)$ entre os usuários em vez de se considerar a razão das somas $\sum B / \sum C$ conforme feito em [Zhu et al. 2018], buscando evitar que apenas alguns usuários sejam privilegiados no processo. No caso da razão das somas, cada componente k da soma recebe pesos diferentes com base no valor C_k. A Equação 22 mostra a diferença dos pesos ao utilizar os dois tipos de média.

$$\frac{\sum B}{\sum C} = \sum \frac{B}{\sum C} = \sum \frac{B}{C} \cdot \frac{C}{\sum C} \neq \sum \frac{B}{C} \cdot 1 = \sum \frac{B}{C} \tag{22}$$

Este fato influencia bastante na recompensa e na forma de seleção de recursos entre usuários, especialmente quando o termo B_k é diferente de zero para mais de um usuário (OFDM). Assim, propõe-se que o custo $C_k(s,a)$ do usuário k seja dado por:

$$C_k(s,a) = P_k(s,a)\sum_{k=1}^{K}(fk) \tag{23}$$

onde f_k é a pressão no *buffer* dada por:

$$f_k = e^{0.5A_k} \tag{24}$$

sendo A_k a quantidade de pacotes no *buffer* [Zhu et al. 2018] do usuário k após ação a que da origem a função de recompensa mdpPltQl1 ou a quantidade de pacotes perdidos do usuário k após ação a que da origem a função de recompensa mdpPltQl2. Ao Considerar uma função exponencial para o termo de pressão ,evita-se a divisão por zero e aumenta-se a diferença entre as recompensas de diferentes estados e ações. Assim, ao explicitar o termo de custo C_k a equação para as funções de utilidade temos:

$$R(s,a) = \sum_{k=1}^{K}\frac{B_k(s,a)}{P_k(s,a)\sum_{k=1}^{K}e^{0.5.A_k}} \tag{25}$$

A função utilidade proposta visa contemplar com maior intensidade cenários onde o *buffer* fica cheio. Ao utilizar os pacotes perdidos como parâmetro (A_k) podemos recompensar de forma diferente com base na quantidade de pacotes perdidos. O horizonte do denominador fica mais amplo, já que A_k não está limitado ao tamanho do *buffer*.

Neste trabalho, avaliaremos a utilização dessas 2 funções de utilidade como funções objetivo no algoritmo de aprendizado por reforço, uma apresentada em [Zhu et al. 2018] e a outra representada pela equação (25). O algoritmo de aprendizado por reforço considerado é baseado no algoritmo *Q-Learning* com iteração de política. Como fazemos uso de um modelo Markoviano para o sistema de comunicação, aqui representado pelas matrizes de probabilidade de transição de estados, chamamos as duas abordagens avaliadas de alocação de recursos de: mdpPltQl1 que corresponde a função proposta em [Zhu et al. 2018] e mdpPltQl2, que propomos, substituindo o tamanho do *buffer* pela quantidade de pacotes perdidos além de considerar o comportamento estatístico dos parâmetros e tomar a média das razões ao invés da razão das médias para Eficiência energética. O termo *MDP (Markov Decision Process)* se refere ao processo de decisão de Markov e o termo *PltQL* se refere ao algoritmo de aprendizado por reforço com iteração política (*off-policy*) utilizado.

Definição 1 *Sejam as variáveis V, j_k, K, $P_k(s,a)$ e l_k, a taxa de codificação, modo atribuído ao usuário k, número de usuários, potência alocada ao usuário k pela ação a quando no estado s e o número de pacotes no buffer do usuário k após a ação a , respectivamente, define-se a função de utilidade Zhu pela seguinte equação:*

$$R_{Zhu}(s,a) = \frac{\sum_{k=1}^{K}V.j_k(a)}{\sum_{k=1}^{K}P_k(s,a)\sum_{k=1}^{K}e^{0.5.l_k(s)}} \tag{26}$$

Proposição 1 *Sejam as variáveis $EE(s,a)$, V, j_k, K, $P_k(s,a)$, $Lost_k^\lambda$ e l_k^λ, a eficiência energética, a taxa de codificação, o modo atribuído ao usuário k, o número de usuários, a potência alocada ao usuário k pela ação a quando no estado s, o número de pacotes perdidos pelo usuário k após a ação a para taxa média de chegada λ e o número de pacotes no buffer do usuário k após da ação a para taxa média de chegada λ, respectivamente, propõem-se as funções de utilidade propostas pelas seguintes equações:*

$$R_{mdpPltQl1}(s,a) = \frac{\frac{1}{K}\sum_{k=1}^{K}\frac{V \cdot j_k(a)}{P_k(s,a)}}{\sum_{k=1}^{K} E[e^{0.5.l_k^\lambda(s)}]} \tag{27}$$

$$R_{mdpPltQl2}(s,a) = \frac{\frac{1}{K}\sum_{k=1}^{K}\frac{V \cdot j_k(a)}{P_k(s,a)}}{\sum_{k=1}^{K} E[e^{0.5.(Lost_k^\lambda(s))}]} \tag{28}$$

onde E[ ] representa o operador esperança.

3.4. Algoritmo Q-Learning

Uma política π é um vetor de ações escolhidas para cada estado s dentre os Ns estados possíveis do sistema, ou seja, é uma realização das $(Na)^{Ns}$ possíveis políticas. Com Na ações possíveis e Ns estados únicos temos $(Na)^{Ns}$ permutações de política. O algoritmo *Q-Learning* utiliza uma função objetivo (que correspondem às denominações mdpPltQl1 - função utilizada [Zhu et al. 2018] mas agora em um cenário OFDM e mdpPltQl2 - Proposta deste artigo) para calcular a recompensa imediata do estado atual s_i ao seguir determinada ação $\pi(s_i)$: r_i^π.

Para um caso real em que não se conhece π utiliza-se a melhor estimativa de π, obtida da otimização do próximo passo. Ou seja, ao tomar a ação ótima para cada passo espera-se chegar a uma política que seja ótima. Essa consideração simplifica o problema com horizonte aparentemente infinito de passos em sub-problemas menores encadeados. A equação (29) é conhecida como equação de Bellman [Sutton and Barto 2018].

$$Q(s_i,a_i) \leftarrow R(s_i,a_i) + \gamma Q(s_{i+1},a_{max}) \tag{29}$$

$$Q(s_i,a_i) \leftarrow R(s_i,a_i) + \gamma \sum_{s'} P(s_i,s',a_{max}) Q(s',a_{max})$$

onde $R(s_i,a_i)$ é a recompensa imediata dada pela função utilidade e

$$a_{max} = Arg_a Max[\sum_{s'} P(s_i,s',a) Q(s',a)] \tag{30}$$

O modelo Markoviano considerado para o sistema de comunicação provê a matriz P dada pela equação (19) e assim é possível encontrar a ação a_{max} para as transições de estado $s_i \rightarrow s_{i+1}$. Basicamente, o algoritmo *Q-Learning* consiste de adaptar o valor de Q e a política ótima π até que π se estabilize ou que se tenha atingido o número máximo de iterações. Ao convergir, a política π composta pela ação a ser tomada em cada estado será dada por:

$$\pi(s_i) = Arg_a Max[Q(s_i,a)] \tag{31}$$

4. Resultados e Discussões

Nessa seção, apresenta-se e discute-se os resultados obtidos com a simulação de um sistema multiportadora com transmissão via ondas milimétricas. Neste trabalho, considerou-se uma frequência de portadora de 6 GHz e uma distância de 80m entre transmissor e receptor. Foram simulados 12 segundos (equivalente a 6000 *frames*). O sistema escolhido varia $K = 1, 2, ..., 6$ usuários, tamanho de *buffer* $L = 2$, quantidade de canais $M = 2$, quantidade de modos de transmissão $J = 3$ e quantidade de estados de canal $C = 2$. Foram simuladas 10 taxas médias de chegada diferentes $\lambda = 0,3; 0,6; ...; 3$ pacotes e posteriormente retirada a média para a plotagem. Para taxa de desconto utilizou-se $\gamma = 0.9$.

Para a modelagem do canal de comunicação, considerou-se que o ganho médio do canal ρ_m é obtido pela média de 100 amostras dos modelos TDL-A, TDL-B, e TDL-C [3GPP 2018]. Como a largura de banda para o sistema considerado é de 20MHz, o valor da potência do ruído é de $10^{-13}W$ para uma densidade de potência do ruído igual a $-100.5dBm$. Para 6 GHz e velocidade do receptor de 1.5 m/s e transmissor fixo tem-se $f_D = \frac{6.10^9}{3.10^8}.1.5 = 30Hz$ para o máximo efeito Doppler, conforme [Rappaport et al. 1996].

A Figura 1 mostra que o algoritmo proposto (Proposta) provê igual ou menor perda de pacotes em relação aos outros algoritmos considerados. Como era de se esperar, observa-se um aumento da perda de pacotes com o aumento do número de usuários no sistema. Ação aleatória corresponde a um algoritmo que selecione aleatoriamente os recursos do sistema OFDM. A ação fixa representa uma escolhe aleatória entre $M = 2$ usuários utilizando os modos com maior capacidade de transmissão (8QAM e 4QAM já que considera-se que não há repetição do modo entre usuários). Pode-se dizer que a menor perda de pacote é reflexo de um processo de tamanho de fila no *buffer* mais limitado. De fato, pode-se observar que se obtém com o algoritmo mdpPltQl2 uma ocupação do *buffer* pela Figura 2 semelhante ao do algoritmo mdpPltQl1, entretanto menor do que o algoritmo baseado em Ação Aleatória.

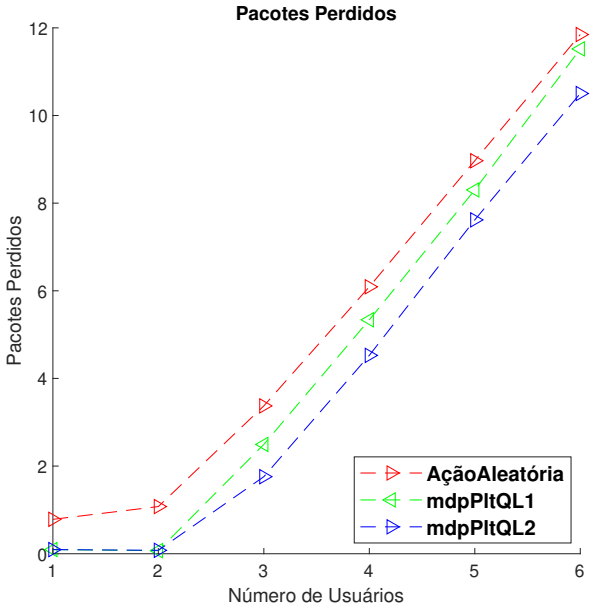

Figura 1. Pacotes perdidos Versus Número de Usuários

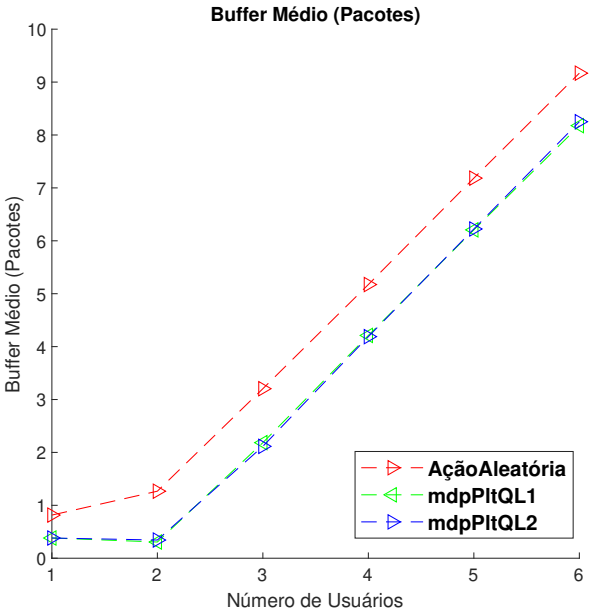

Figura 2. Tamanho da Fila no Buffer Versus Número de Usuários

Com uma menor perda percentual de pacotes e uma estratégia mais eficiente, pode-se afirmar que o algoritmo proposto provê uma maior taxa de transmissão, ou seja, transmite-se mais pacotes conforme mostra a Figura 3. Observa-se que o número de pacotes transmitidos tende a se tornar mais constante com o aumento do número de usuários no sistema.

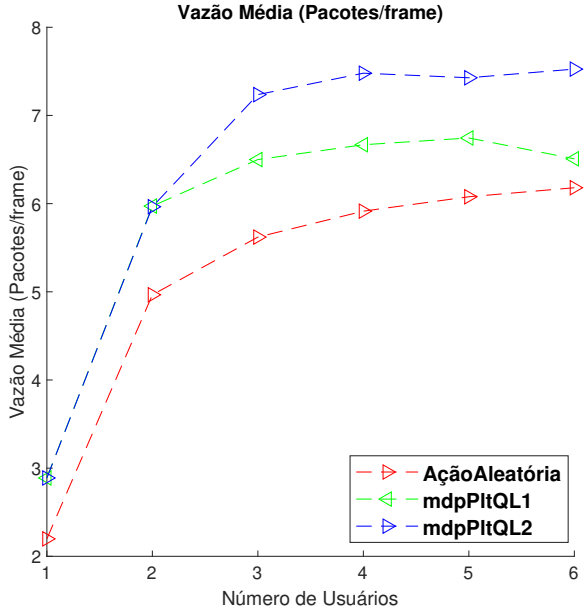

Figura 3. Pacotes transmitidos Versus Número de Usuários

Pode-se observar também que o algoritmo proposto apresenta uma maior eficiência energética mostrada pela Figura 4 que os demais algoritmos considerados. O

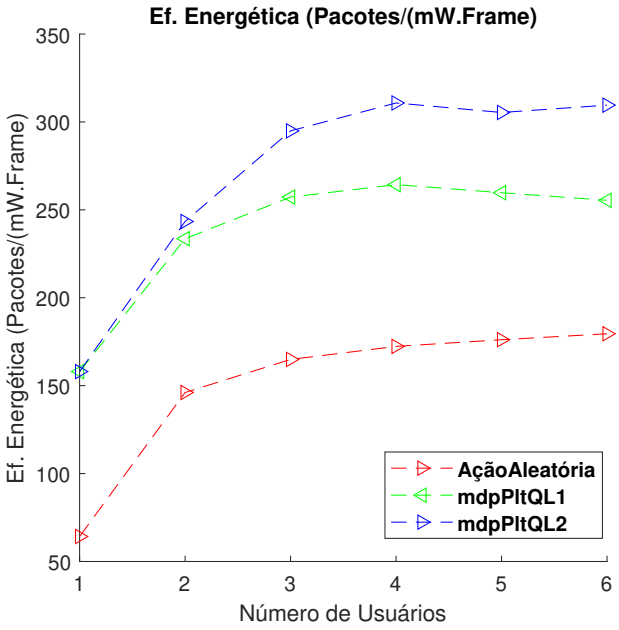

Figura 4. Eficiência Energética Versus Número de Usuários

cálculo da eficiência energética utilizado é dado pela seguinte equação:

$$EE(s,a) = \frac{1}{K}\sum_{k=1}^{K}\frac{B_k(s,a)}{P_k(s,a)} \tag{32}$$

Para corroborar o resultado de eficiência energética do algoritmo proposto, apresentamos na Figura 5 também a potência média alocada por *frame* versus o número de usuários. Observa-se que os algoritmos mdpPltQl1 e mdpPltQl2 apresentam potências médias alocadas semelhantes, mas menores do que a seleção aleatória.

5. Conclusões

Neste artigo, considera-se que um sistema de comunicação OFDM pode ser descrito por um modelo Markoviano e consequentemente um algoritmo de alocação de recursos baseado em aprendizado por reforço pode ser aplicado para escalonamento de recursos. Cada canal pode assumir estados diferentes (sub-portadoras) respeitando uma distribuição de *Rayleigh* para os desvanecimentos impostos ao sinal.

Propôs-se uma função recompensa para o algoritmo baseado em aprendizado por reforço tendo como um dos objetivos minimizar a quantidade de pacotes perdidos explicitando esse parâmetro na função. Foi considerada também a função recompensa utilizada em [Zhu et al. 2018] que não explicita a quantidade de pacotes perdidos diretamente.

Observou-se que as funções de recompensa podem prover resultados distintos. O uso explícito da quantidade de pacotes perdidos na função contribui para reduzir perda de pacotes, aumentar a taxa de transmissão e melhorar a eficiência energética do sistema.

Os parâmetros utilizados para tamanho máximo do *buffer* ($L = 2$ pacotes) e taxa média de chegada de pacotes ($\lambda = 3$ pacotes/intervalo) favorecem a proposta uma vez

que $\lambda > L$. Assim, há maior ocorrência de perda de pacotes que em um cenário onde $\lambda < L$. . Em resumo, se o parâmetro *buffer* fosse de tamanho infinito $Buff_{inf}$, este seria representado pelos termos $Buff + Lost$, ou seja, $Buff_{inf} = Buff + Lost$ onde $Buff$ representa um *buffer* finito de tamanho máximo L e $Lost$ a quantidade de pacotes perdidos com *buffer* finito e de tamanho L. Ao comparar as duas soluções explicita-se a contribuição de cada componente de um *buffer* infinito para a tomada de decisão do agente. Cabe destacar que utilizar a combinação dos dois parâmetros provê resultados mais robustos à variação na relação λ/L.

Por fim, observa-se que algoritmos baseados em aprendizado por reforço podem prover melhoria de desempenho para sistemas de comunicação e que a escolha da função utilidade pode influenciar nas soluções obtidas.

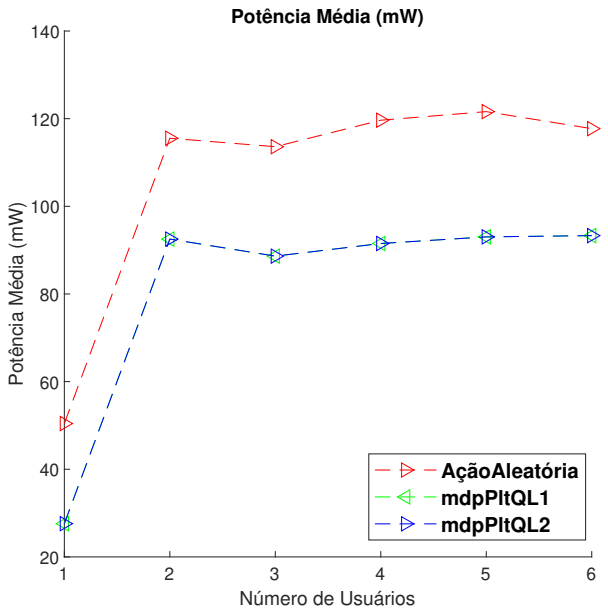

Figura 5. Potência média por frame Versus Número de Usuários

Referências

3GPP (2018). Study on channel model for frequencies from 0.5 to 100 ghz (release 15). Technical report, 3GPP TR 38.901.

Ford, R., Rangan, S., Mellios, E., Kong, D., and Nix, A. (2017). Markov channel-based performance analysis for millimeter wave mobile networks. In *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6. IEEE.

Hong Shen Wang and Moayeri, N. (1995). Finite-state markov channel-a useful model for radio communication channels. *IEEE Transactions on Vehicular Technology*, 44(1):163–171.

Matz, G. and Hlawatsch, F. (2011). Fundamentals of time-varying communication channels. In *Wireless Communications Over Rapidly Time-Varying Channels*, pages 1–63. Elsevier.

Patteti, K., Kumar, T., and Kalitkar, K. (2016). M-qam ber and ser analysis of multipath fading channels in long term evolutions (lte). *International Journal of Signal Processing, Image Processing and Pattern Recognition(IJSIP)*, Vol.9:361–368.

Proakis, J. and Salehi, M. (2008). *Digital Communications*. McGraw-Hill International Edition. McGraw-Hill.

Rappaport, T. S. et al. (1996). *Wireless communications: principles and practice*, volume 2. prentice hall PTR New Jersey.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Zhu, J., Song, Y., Jiang, D., and Song, H. (2018). A new deep-q-learning-based transmission scheduling mechanism for the cognitive internet of things. *IEEE Internet of Things Journal*, 5(4):2375–2385.