Escalonamento de Recursos em Sistema Multiportadora com Ondas Milimétricas Utilizando Aprendizado por Reforço Markoviano

Daniel Porto Queiroz Carneiro
Escola de Engenharia Elétrica, Mecânica e de Computação - EMC
3º piso, Av. Universitária, 1488 - Qd. 86 Bloco A - Setor Leste Universitário, Goiânia - GO, 74605-010
danpqcar@gmail.com

Álisson Assis Cardoso
Escola de Engenharia Elétrica, Mecânica e de Computação - EMC
3º piso, Av. Universitária, 1488 - Qd. 86 Bloco A - Setor Leste Universitário, Goiânia - GO, 74605-010
alsnac@gmail.com

Flávio Henrique Teles Vieira
Escola de Engenharia Elétrica, Mecânica e de Computação - EMC
3º piso, Av. Universitária, 1488 - Qd. 86 Bloco A - Setor Leste Universitário, Goiânia - GO, 74605-010
flaviohtv@gmail.com

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 nova função de recompensa utilizada no 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 variações do algoritmo *Q-learning* presentes na literatura são realizadas, mostrando também que o uso da função de recompensa proposta torna o escalonamento de usuários e o compartilhamento de recursos mais eficientes.

PALAVRAS CHAVE. Aprendizado por Reforço, Sistema Multiportadora, Canais variantes no tempo.

Modelagem markoviana dos estados do *buffer* e canal, aprendizado por reforço

ABSTRACT

In this article, 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 introduced a new reward functions 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 variations of the Q-learning algorithm present in the literature are carried out, also showing that the use of the proposed reward function makes user scheduling and resource sharing more efficient.

KEYWORDS. Reinforcement Learning, Multicarrier Systems, time varying channels.

Markovian modeling of the buffer and channel states, reinforcement learning

1. Introdução

Um dos desafios em sistemas de comunicação é compartilhar recursos de forma eficiente sendo estes limitados. Além do número de frequências disponíveis de transmissão ser finito, 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. 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 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 de [Ford et al., 2017], aborda-se o desempenho de um sistema LTE OFDM (*Long Term Evolution Orthogonal Frequency 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 para multiusuários 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* - Probabilidade 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 aleatório [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 por Hong Shen Wang e 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 direta. 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. Algoritmo de aprendizado por reforço adaptado para otimizar parâmetros de qualidade de serviço em um sistema de comunicação OFDM multiusuários;

2. Proposta de função de utilidade para o algoritmo de aprendizagem por reforço;

3. 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, 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 e 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 $v^2 = P \cdot |h|^2$ do sinal transmitido é, portanto, exponencial [Proakis e Salehi, 2008]. Assumindo que a P (Potência na estação base) e WN_0 sejam constantes no *frame* (2 ms), a SNR (*Signal to Noise Ratio*) 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 frenquência 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. Para valores de ganho abaixo de um certo valor, o atendimento para o usuário não é realizado (estado ocioso).

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 4 modos diferentes de transmissão BPSK, 4QAM, 8QAM e 16QAM proporcionando diferentes taxas de geração de pacotes. O valor de potência a ser alocada é obtido explicitando o termo P nas equações de probabilidade de erro de bit ($pBER$) referentes às modulações consideradas, dadas a seguir:

- $BPSK, j = 1$

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

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

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

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 e 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 TDM

Essa ação atende um único usuário de cada vez, escolhe-se em que canal transmitir, qual usuário e qual modo de transmissão, ou seja $K \cdot M \cdot J + 1$ ações diferentes. Estas ações se referem ao cenário em que atende-se um usuário de cada vez no *frame* com ações TDM (*Time-Division Multiplexing*) , que chamaremos de Cenário 1.

3.2. 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, que chamaremos de Cenário 2 neste artigo.

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:

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

3.3. 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 $S_c = C_h^M$ 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.

3.3.1. 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* é:

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

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.

3.3.2. Estados do canal

O ambiente simulado possui Ch estados possíveis para cada um dos M canais $C = c_0, ...c_{Ch-1}$ de acordo com o ganho $|h|^2$ do canal e $Ch - 1$ limiares. $\rho = \rho_1, \rho_2, ...\rho_{Ch-1}$, com $\rho_0 = 0$ e $\rho_{Ch} = inf$. 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), obtem-se:

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

Adota-se modelo markoviano também para os estados dos canais. A transição de estado dos canais se dá apenas entre estados vizinhos da cadeia. Considerando um processo Markoviano de nascimento e morte, as probabilidades de transição de um estado com melhor ou pior canal ocorrem de forma independente para cada canal e são dadas pelas seguintes equações:

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

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

onde Tf é a duração do *frame* em segundos e $N(c_n)$ é o numero de vezes que o limiar de c_n é cruzado por segundo, ou seja, o numerador nas equações (15) e (16) são estimativas para a probabilidade do canal mudar de estado em 1 *frame*. Ao se dividir os numeradores das equações (15) e (16) por $pc(c_n)$ obtém-se probabilidades condicionais, i.e., probabilidades de determinadas transições ocorrerem dado que o estado atual (c_n) é conhecido. O valor de $N(c_n)$ é obtido utilizando a seguinte equação [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 Ch estados possíveis e pode-se utilizar mais de um canal ao mesmo tempo com o atendimento OFDM, tem-se Ch^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 Ch^M estados, e $c(m)$ e $c(m)^{'}$ o estado individual de cada canal, do conjunto de Ch 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 é:

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

3.4. Função Utilidade

A função utilidade é responsável por agrupar as variáveis do sistema e modelar uma relação entre elas que permita à 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:

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

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

onde l_i 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.

Neste trabalho, propõe-se adotar na função utilidade, 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. Este fato influencia bastante na recompensa e no nível de justiça (*fairness*) na seleção de recursos entre usuários. Assim, propõe-se que o custo $C_k(s,a)$ do usuário k seja dado por:

$$C_k(s,a) = P_k(s,a) \sum_{k=1}^{K} (fk) \tag{22}$$

onde f_k é a pressão no *buffer*.

$$fk = e^{0.5A_k} \tag{23}$$

onde A_k é a quantidade de pacotes no *buffer* [Zhu et al., 2018] ou a quantidade de pacotes perdidos do usuário k após ação a. Considera-se uma função exponencial à pressão para evitar a divisão por zero e ajudar a diferenciar os possíveis valores do expoente. Assim, a equação para a função utilidade proposta se torna:

$$R(s,a) = \sum_{k=1}^{K} \frac{B_k(s,a)}{P_k(s,a) \sum_{k=1}^{K} e^{0.5.A_k}} \tag{24}$$

A função utilidade proposta visa contemplar com maior intensidade cenários onde o *buffer* fica cheio. Ao utilizar os pacotes perdidos como parâmetros 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*. Para tamanho de *buffer* (L) cuja ordem de grandeza é maior do que a taxa λ de chegada, tem-se da equação (9) que $Prob(L,\lambda) \approx 0$, ou seja, as funções de recompensa proposta e a apresentada em [Zhu et al., 2018] se aproximam.

Neste trabalho, avaliaremos a utilização dessas 2 funções de utilidade como funções objetivo no algoritmo de aprendizado por reforço. O algoritmo de aprendizado por reforço considerado é baseado no algoritmo *Q-Learning* com iteração de política.

3.5. 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 serão denominadas como função 1 - Zhu [Zhu et al., 2018] e função 2 - Proposta) 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 (25) é conhecida como equação de Bellman [Sutton e Barto, 2018].

$$Q(s_i, a_i) \leftarrow R(s_i, a_i) + \gamma Q(s_{i+1}, a_{max}) \tag{25}$$

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

onde $R(s_i, a_i)$ é a recompensa imediata e

$$a_{max} = Arg_a Max[\sum_{s'} P(s_i, s', a) Q(s', a)] \tag{26}$$

O modelo Markoviano considerado para o sistema de comunicação provê a matriz P dada por (21) 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:

$$\pi(s_i) = Arg_a Max[Q(s_i, a)] \tag{27}$$

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 20 segundos (equivalente a 10000 *frames*). O sistema escolhido tem K=3 usuários, tamanho de *buffer* $L = 2$, quantidade de canais $M = 3$, quantidade de modos de transmissão $J = 4$ e quantidade de estados de canal $C_h = 4$. A taxa de chegada durante a simulação foi variada entre 0.1 e 11.5 ($\lambda = [0.1, 0.7, 1.3, ..., 11.5]$). Para taxa de desconto utilizou-se $\gamma = 0.9$. 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]. Durante a simulação, para ganhos de canal menores que $0.01\frac{\rho_m}{10^{PL/10}}$ para os dispositivos de usuários, considera-se condição de inoperância, ou seja, o atendimento à demanda do usuário não é realizada. 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 [Rappaport et al., 1996].

Foram obtidos resultados para 2 cenários de simulação. O cenário 1 contempla ações individuais (TDM). São comparadas as duas funções utilidade para esse cenário. O cenário 2 contempla as ações completas incluindo além das ações simples (TDM), as ações OFDM. Para diferenciar os

resultados dos diferentes cenários e as funções é adotada a nomenclatura: Cen1-ZhuQL, Cen1-Proposta, Cen2-ZhuQL e Cen2-Proposta. Além da aplicação de escalonamento de recursos via aprendizagem por reforço, considerou-se também nas simulações uma seleção aleatória de recursos tanto para o cenário 1 como para o cenário 2.

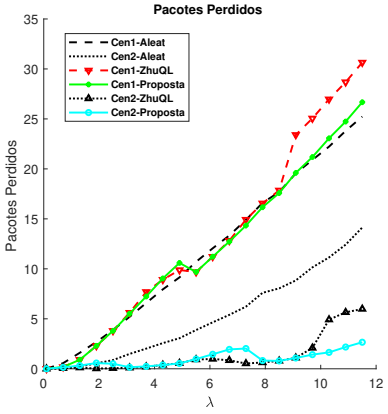

Figura 1: Pacotes perdidos versus taxa média de geração de pacotes

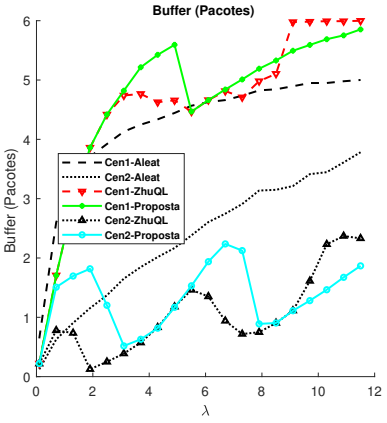

Figura 2: Tamanho do Buffer versus taxa média de geração de pacotes

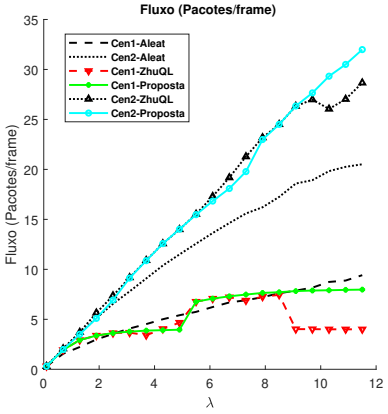

Figura 3: Pacotes transmitidos versus taxa média de geração de pacotes

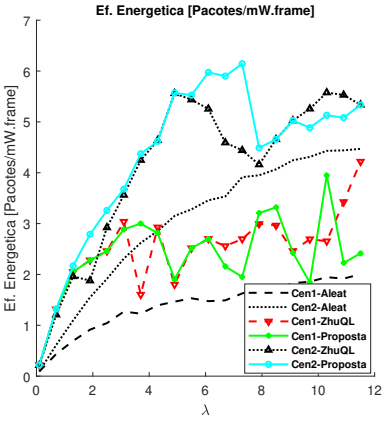

Figura 4: Eficiência Energética versus taxa média de geração de pacotes

No cenário 1 para valores de $\lambda > 8.1$, o algoritmo proposto provê igual ou menor perda de pacotes (Fig. 1) em relação aos outros algoritmos considerados. Além disso, para $\lambda > 8.1$ pode-se observar que se obtém com o algoritmo proposto uma menor ocupação do *buffer* pela Fig. 2, transmite-se mais pacotes conforme mostra a Figura 3. O cálculo da eficiência energética utilizado é dado pela seguinte equação:

$$EE(s,a) = \frac{1}{K}\sum_{k=1}^{K}\frac{B_k(s,a)}{P_k(s,a)} \tag{28}$$

No cenário 2 vê-se maior diferença no comportamento das duas soluções (especialmente quanto aos pacotes perdidos e transmitidos) para λ maior que 8.1. Em média, menos pacotes são perdidos para

faixa de valores de λ (Fig. 1), há menor ocupação do *buffer* (Fig. 2), mais pacotes são transmitidos (Fig. 3).

Na Figura 4 pode-se visualizar a vantagem dos algoritmos propostos na equação 20 em relação a alocação aleatória quanto a eficiência energética. Porém, ao se comparar as propostas, não é possível destacar o desempenho de uma em relação a outra (Fig. 4).

Quando se deseja o melhor desempenho possível para o sistema, avalia-se questões como menor espera dos pacotes na fila (menor tamanho de fila no *buffer*), menor perda de pacotes e maiores taxas. O algoritmo proposto se mostrou superior nesses requisitos para taxa média de chegada acima de 8.1 pacotes por *frame*. Como era de se esperar, os algoritmos baseados em aprendizagem por reforço foram mais eficientes do que a alocação aleatória de recursos para o cenário 2 em relação a todos os parâmetros de desempenho analisados.

Segundo os critérios estabelecidos de eficiência energética e parâmetros como fluxo, perda de pacotes, tamanho da fila no *buffer*, o uso de OFDM+TDM (cenário 2) mostra maior vantagem quando comparado ao modo TDM simples (cenário 1), apesar da maior complexidade do cenário 2. As diferenças nos indicadores de QoS entre os 2 cenários ocorrem quando as taxas de chegada são grandes $\lambda > 0.7$ (pelo menos em comparação com o tamanho da fila no *buffer*). Para taxas $\lambda < 1.3$ a função de recompensa (de [Zhu et al., 2018] e a proposta) no cenário 2 aponta para políticas que não utilizam ações OFDM. Há semelhança dos gráficos de Qos para $\lambda < 1.3$ para cenários 1 e 2. São situações em que não se perde pacotes mesmo atendendo um usuário de cada vez devido a baixa taxa de chegada de dados. A medida que a taxa de chegada aumenta, nota-se as curvas dos cenários 1 e 2 se afastarem.

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 por Zhu et al. [2018] que não explicita perda de pacotes diretamente.

Observou-se que as duas propostas de função de recompensa podem prover resultados distintos. O uso explícito da quantidade de pacotes perdidos na função contribui para reduzir perda de pacotes e aumentar a taxa de transmissão quando a taxa média de chegada é superior a 8 pacotes por frame (2 ms).

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.

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., e Nix, A. (2017). Markov channel-based performance analysis for millimeter wave mobile networks. In *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, p. 1–6. IEEE.

Hong Shen Wang e 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. e Hlawatsch, F. (2011). Fundamentals of time-varying communication channels. In *Wireless Communications Over Rapidly Time-Varying Channels*, p. 1–63. Elsevier.

Patteti, K., Kumar, T., e 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. e Salehi, M. (2008). *Digital Communications*. McGraw-Hill International Edition. McGraw-Hill. ISBN 9780071263788.

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

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

Zhu, J., Song, Y., Jiang, D., e 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.