UNIVERSIDADE FEDERAL DE GOIÁS (UFG)

ESCOLA DE ENGENHARIA ELÉTRICA, MECÂNICA E DE COMPUTAÇÃO

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO

MATHEUS MATOS VASCONCELOS

# Alocação de Recursos em Sistemas Internet das Coisas Utilizando Aprendizagem por Reforço

Goiânia
2021

UNIVERSIDADE FEDERAL DE GOIÁS
ESCOLA DE ENGENHARIA ELÉTRICA, MECÂNICA E DE COMPUTAÇÃO

# TERMO DE CIÊNCIA E DE AUTORIZAÇÃO (TECA) PARA DISPONIBILIZAR VERSÕES ELETRÔNICAS DE TESES

# E DISSERTAÇÕES NA BIBLIOTECA DIGITAL DA UFG

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio da Biblioteca Digital de Teses e Dissertações (BDTD/UFG), regulamentada pela Resolução CEPEC nº 832/2007, sem ressarcimento dos direitos autorais, de acordo com a Lei 9.610/98, o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou download, a título de divulgação da produção científica brasileira, a partir desta data.

O conteúdo das Teses e Dissertações disponibilizado na BDTD/UFG é de responsabilidade exclusiva do autor. Ao encaminhar o produto final, o autor(a) e o(a) orientador(a) firmam o compromisso de que o trabalho não contém nenhuma violação de quaisquer direitos autorais ou outro direito de terceiros.

## 1. Identificação do material bibliográfico

[ X ] Dissertação [ ] Tese

## 2. Nome completo do autor

Matheus Matos Vasconcelos

## 3. Título do trabalho

**“Alocação de Recursos em Sistemas Internet das Coisas Utilizando Aprendizagem por Reforço”**

## 4. Informações de acesso ao documento (este campo deve ser preenchido pelo orientador)

Concordа com a liberação total do documento [ X ] SIM [ ] NÃO[1]

**[1]** Neste caso o documento será embargado por até um ano a partir da data de defesa. Após esse período, a possível disponibilização ocorrerá apenas mediante:
**a)** consulta ao(à) autor(a) e ao(à) orientador(a);
**b)** novo Termo de Ciência e de Autorização (TECA) assinado e inserido no arquivo da tese ou dissertação.
O documento não será disponibilizado durante o período de embargo.
Casos de embargo:
- Solicitação de registro de patente;
- Submissão de artigo em revista científica;
- Publicação como capítulo de livro;
- Publicação da dissertação/tese em livro.

**Obs. Este termo deverá ser assinado no SEI pelo orientador e pelo autor.**

Documento assinado eletronicamente por **Flavio Henrique Teles Vieira**, **Professor do Magistério Superior**, em 12/11/2021, às 17:29, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **MATHEUS MATOS VASCONCELOS**, **Discente**, em 12/11/2021, às 17:59, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade deste documento pode ser conferida no site https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **2488529** e o código CRC **49B06B4B**.

**Referência:** Processo nº 23070.038177/2021-93 SEI nº 2488529

MATHEUS MATOS VASCONCELOS

# Alocação de Recursos em Sistemas Internet das Coisas Utilizando Aprendizagem por Reforço

Dissertação apresentada ao Programa de Pós–Graduação do Escola de Engenharia Elétrica, Mecânica e de Computação da Universidade Federal de Goiás (UFG), como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica e de Computação.

**Área de concentração:** Engenharia de Computação.

**Orientador:** Prof. Dr. Flávio Henrique Teles Vieira

**Co-Orientador:** Prof. Dr. Álisson Assis Cardoso

Goiânia
2021

Ficha de identificação da obra elaborada pelo autor, através do
Programa de Geração Automática do Sistema de Bibliotecas da UFG.

Vasconcelos, Matheus Matos
Alocação de Recursos em Sistemas Internet das Coisas Utilizando Aprendizagem por Reforço [manuscrito] / Matheus Matos Vasconcelos. - 2021.
51 f.: il.

Orientador: Prof. Dr. Flávio Henrique Teles Vieira; co-orientador Dr. Álisson Assis Cardoso.
Dissertação (Mestrado) - Universidade Federal de Goiás, Escola de Engenharia Elétrica, Mecânica e de Computação (EMC), Programa de Pós-Graduação em Engenharia Elétrica e de Computação, Goiânia, 2021.
Bibliografia.

1. Aprendizagem por Reforço. 2. Cadeia de Markov. 3. Escalonamento. I. Vieira, Flávio Henrique Teles, orient. II. Título.

CDU 621.3

UNIVERSIDADE FEDERAL DE GOIÁS

ESCOLA DE ENGENHARIA ELÉTRICA, MECÂNICA E DE COMPUTAÇÃO

# ATA DE DEFESA DE DISSERTAÇÃO

Ata nº 07 da sessão de Defesa de Dissertação de **Matheus Matos Vasconcelos**, que confere o título de Mestre em **Engenharia Elétrica e de Computação**, na área de concentração em **Engenharia de Computação**.

Aos **quatro dias do mês de agosto de dois mil e vinte um**, a partir das **14h00min**, realizou-se a sessão pública de Defesa de Dissertação intitulada **"Alocação de Recursos em Sistemas Internet das Coisas Utilizando Aprendizagem por Reforço"**. Os trabalhos foram instalados pelo Orientador, Professor Doutor **Flávio Henrique Teles Vieira (EMC/UFG)**, com a participação dos demais membros da Banca Examinadora: Professor Doutor **Kleber Vieira Cardoso (INF/UFG)** membro titular externo; Professor Doutor **Flávio Geraldo Coelho Rocha (EMC/UFG)** membro titular interno; Professor Doutor **Alisson Assis Cardoso (PPGEEC-UFG) -** Co-orientador**: cujas participações ocorreram através de videoconferência**. Durante a arguição os membros da banca **não fizeram** sugestão de alteração do título do trabalho**.** A Banca Examinadora reuniu-se em sessão secreta a fim de concluir o julgamento da Dissertação, tendo sido o candidato **aprovado** pelos seus membros. Proclamados os resultados pelo Professor Doutor **Flávio Henrique Teles Vieira**, Presidente da Banca Examinadora, foram encerrados os trabalhos e, para constar, lavrou-se a presente ata que é assinada pelos Membros da Banca Examinadora, aos **quatro dias do mês de agosto de dois mil e vinte um**.

TÍTULO SUGERIDO PELA BANCA

Documento assinado eletronicamente por **Flávio Geraldo Coelho Rocha**, **Professor do Magistério Superior**, em 04/08/2021, às 15:52, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Flavio Henrique Teles Vieira**, **Professor do Magistério Superior**, em 04/08/2021, às 15:52, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **ALISSON ASSIS CARDOSO**, **Usuário Externo**, em 04/08/2021, às 15:54, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Kleber Vieira Cardoso**, **Professor do Magistério Superior**, em 04/08/2021, às 15:56, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **MATHEUS MATOS**

**VASCONCELOS**, **Discente**, em 04/08/2021, às 16:50, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **2244318** e o código CRC **EA5A9FD4**.

**Referência:** Processo nº 23070.038177/2021-93 SEI nº 2244318

Matheus Matos Vasconcelos

# Alocação de Recursos em Sistemas Internet das Coisas Utilizando Aprendizagem por Reforço

Dissertação apresentada ao Programa de Pós–Graduação doEscola de Engenharia Elétrica, Mecânica e de Computaçãoda Universidade Federal de Goiás (UFG), como requisito parcial para obtenção do título de Mestre em EngenhariaElétrica e de Computação.

---

**Prof. Dr. Flávio Henrique Teles Vieira**
Orientador

---

**Prof. Dr. Álisson Assis Cardoso**
Coorientador

---

**Prof. Dr. Kleber Vieira Cardoso**
Convidado 1

---

**Prof. Dr. Flávio Geraldo Coelho Rocha**
Convidado 2

Goiânia
2021

# Resumo

Este trabalho propõe a utilização de um algoritmo de aprendizagem por reforço (AR) para controlar a transmissão de pacotes de múltiplos dispositivos em um sistema de comunicação sem fio baseado no conceito de Internet das Coisas (IdC) Cognitivo. A abordagem proposta consiste em adotar uma cadeia de Markov para modelar os estados do sistema de comunicação e suas transições, fornecendo os parâmetros necessários para determinar ações para o sistema através de um algoritmo *Q-Learning*. O trabalho contém uma avaliação do desempenho do algoritmo desenvolvido em comparação aos de alguns algoritmos de escalonamento conhecidos na literatura em termos de vários parâmetros, tais como: função de utilidade, vazão, ocupação do *buffer*, taxa de perda de pacotes, etc.

**Palavras-chave**: Aprendizagem por Reforço. Cadeia de Markov. Escalonamento. Internet das Coisas (IdC).

# Abstract

This paper proposes a utilization of a reinforcement learning (RL) algorithm to control the packet transmission of multiple devices of a Cognitive Internet of Things (IoT) wireless communication system. The proposed approach consists of adopting a Markov chain to model the states of the communication system and its transitions, providing the required parameters to determine actions to the system using a Q-Learning algorithm. This paper also presents a performance evaluation of the developed algorithm in comparison to some scheduling algorithms in terms of: utility function, flow rate, buffer occupancy, packet loss rate, etc.

**Keywords**: Internet of Things (IoT). Markov Chain. Reinforcement Learning. Scheduling.

# Trabalhos Submetidos e Publicados

**Trabalhos aprovados e/ou publicados**:

- VASCONCELOS, M. M. ; CARDOSO A. A. ; VIEIRA , F. H. T. . Algoritmo Baseado em Aprendizado por Reforço e Modelo Markoviano para Alocação de Recursos em um Sistema Internet das Coisas Cognitivo. In: *XXXVIII Simpósio Brasileiro de Telecomunicações e Processamento de Sinais - SBrT 2020*. Florianópolis, Santa Catarina, 2020.
- VASCONCELOS, M. M. ; CARDOSO A. A. ; VIEIRA , F. H. T. . Alocação de Recursos em Sistemas Internet das Coisas Utilizando Aprendizagem por Reforço e Modelo Markoviano. In: *X Conferência Nacional em Comunicações, Redes e Segurança da Informação - ENCOM 2020*. Natal, Rio Grande do Norte

# Sumário

# Introdução

Observa-se um crescente aumento no uso da Inteligência Artificial (IA) em diferentes ramos da ciência uma vez que os problemas vêm se tornando cada vez mais complexos e demandando soluções menos restritivas que se adaptem a natureza não trivial das dificuldades modernas (JANG et al., 2019; XIONG et al., 2019). Nesses cenários, o emprego de IA é ainda mais desejável visto que a mesma possibilita que sistemas sejam capazes de aprender e tomar decisões onde não há soluções ótimas claras.

Problemas de alocação de recursos podem ser complexos a ponto de não ser possível encontrar a solução ótima por alguns algoritmos em tempo razoável. Nesse sentido, algoritmos baseados em inteligência computacional podem ajudar a encontrar uma solução satisfatória pois podem facilitar o tratamento de grandes quantidades de dados, podem aumentar a velocidade de análise e permitem que processos complexos sejam automatizados (JANG et al., 2019; ZHU et al., 2019). Aumenta-se então o interesse e a possibilidade de uso de IA como uma ferramenta de coordenação de ações em cenários IdC cognitivos, onde dispositivos e sensores inteligentes exigem uma certa flexibilidade do sistema por conta da dinâmica de integração, resultando em cenários mais complexos e com uma ampla variedade de aplicações.

Algoritmos baseados em técnicas de aprendizagem de máquina possuem capacidade de aprender e se adaptar, permitindo a otimização dos recursos do sistema. No caso de um cenário IdC cognitivo, por exemplo, a aplicação de tais algoritmos pode proporcionar aumento de vazão de dados enquanto se procura reduzir o custo total de transmissão, melhorando assim a qualidade de serviço e a eficiência do sistema (ZHU et al., 2019; XIONG et al., 2019). Na aprendizagem por reforço são avaliadas as ações possíveis a serem tomadas, permitindo determinar um curso de ações para cada estado do sistema levando em consideração as recompensas obtidas para cada ação.

Este trabalho está dividido da seguinte forma: no Capítulo 1 são apresentados os objetivos deste trabalho; no Capítulo 2 são abordados conceitos teóricos necessários para uma boa compreensão deste trabalho; no Capítulo 3 é descrita a modelagem do sistema IdC por meio de um modelo Markoviano bem como as considerações para os cálculos dos parâmetros necessários para esta modelagem para realizar as simulações apresentadas no Capítulo 4. É introduzido também no Capítulo 3 o algoritmo proposto que utiliza aprendizagem por reforço para o treinamento do agente; no Capítulo 4 são apresentados os valores dos parâmetros e as considerações utilizadas nas simulações e são mostrados também os resultados dos algoritmos considerados; finalmente, no Capítulo 5 são apresentadas as conclusões acerca do trabalho.

# 1 Objetivos

Neste trabalho, propõe-se a utilização de um algoritmo de aprendizagem por reforço que faz uso de um modelo de Markov para descrever os estados de um sistema IdC cognitivo através das probabilidades de transições dos estados e determinar uma política de ações que uma estação base deverá tomar para aumentar um parâmetro chamado utilidade do sistema. Este parâmetro provê informação acerca de três características de um sistema de transmissão de pacotes, a vazão de pacotes, a potência consumida e o uso do *buffer*.

O sistema IdC considerado consiste em uma rede de dispositivos que recebem pacotes de dados (*downlink*) e os enviam para uma estação base (*uplink*). Esta estação base possui informações acerca do estado do *buffer* dos seus dispositivos e, a partir dessas informações, é capaz de fazer inferências e tomar decisões para alocar os seus recursos e fazer o gerenciamento de quais dispositivos podem transmitir seus pacotes em cada instante de tempo. Em razão do sistema ser capaz de observar, analisar e tomar decisões, o sistema é considerado cognitivo.

Este trabalho também faz uma comparação do desempenho do algoritmo proposto com o de outros que possuem funções similares conhecidas na literatura, para avaliar seu desempenho quanto as três características de interesse da utilidade do sistema.

Em outras palavras, a abordagem proposta objetiva aumentar a utilidade do sistema representada pelo valor da quantidade de pacotes transmitidos, pela potência consumida para a transmissão e por um parâmetro que identifica o uso do *buffer*.

# 2 Referencial Teórico

Este capítulo apresenta os conceitos teóricos envolvendo Internet das Coisas e aprendizagem por reforço utilizados na elaboração deste trabalho.

## 2.1 Internet das Coisas

IdC é um sistema de comunicação que permite a conectividade de vários dispositivos sem fio. Esses tipos de sistemas têm sido utilizados em diversas áreas como segurança, transporte e administração como uma forma de fornecer um estilo de vida mais inteligente, mais fácil e mais seguro (Wei et al., 2020).

Sistemas IdC ainda enfrentam diversos desafios quando são usados em cenários práticos. A eficiência do trabalho de controle vai se deteriorando com o tempo caso não haja intervenção humana nos casos de interações e eventos inesperados. Portanto, é muito interessante sistemas IdC com autonomia que reduz a necessidade de assistência humana. Outro desafio é a velocidade de se processar e analisar grandes quantidades de informações, e isso pode causar problemas em sistemas que precisam de um tempo de resposta rápido. Sistemas IdC precisam ter não só habilidades de controle e gestão, como também garantir rápida velocidade de processamento e baixo tempo de resposta.

Essa forma de comunicação entre esses dispositivos também passa por desafios como limitações de memória, de espaço de armazenamento, de consumo de energia que muitas vezes possuem limitações de *hardware* dentro dos cenários em que os dispositivos usados como sensores, receptores, transmissores se encontram (CIRANI et al., 2018).

Para informações serem compartilhadas entre os dispositivos é necessário uma arquitetura de rede que proveja os requisitos do sistema. A tecnologia 5G é uma boa alternativa para integrar sistemas IdC isolados e independentes que usam sistemas de comunicação sem fio de curto alcance, como *BlueTooth* e *Ultrawideband* (UWB), pois o uso do 5G permite uma conexão contínua entre esses sistemas combinados, com capacidade superior ao 4G, maior taxa de dados, menor latência e uma segurança de transmissão de dados melhorada (KHURPADE; RAO; SANGHAVI, 2018).

Sistemas IdC cognitivo consistem numa estação base que precisa conhecer os estados dos seus sensores e dispositivos, possuem ampla aplicação em ambientes com tecnologia *Ambient Assisted Living* (AAL) que funcionam no monitoramento em tempo real de pacientes em hospitais ou em moradias adaptadas para ajudar nas tarefas de idosos ou pessoas que precisam de necessidades especiais (TRIPATHY; ANURADHA, 2017). Esses sistemas mais complexos exigem um maior cuidado com o consumo energético e uma

maior otimização na comunicação digital para evitar perdas de informações que podem ser críticas em cenários reais.

## 2.2 Cadeia de Markov

Por definição, uma cadeia de Markov é um processo markoviano composto de estados finitos no qual o estado futuro do processo, dado um estado atual, depende apenas do estado atual e não dos estados anteriores (PAPOULIS; PILLAI, 2002).

Numa cadeia de Markov com o tempo discreto e com um número finito de estados as probabilidades de transição de estados são representadas por $p_{ij}(t_n, t_m)$, significando a probabilidade do sistema estar no estado $i$ no instante $t_n$ e transicionar para o estado $j$ no instante $t_m$. Para casos onde o instante posterior é exatamente um incremento adicional do instante presente a representação é de apenas $p_{ij}(t_n, t_n + 1) = p_{i,j}$. Um exemplo de uma cadeia de Markov de estados finitos com 3 estados pode ser encontrado na Figura 2.1.

É comum representar as probabilidades de transição de estados em um formato matricial como:

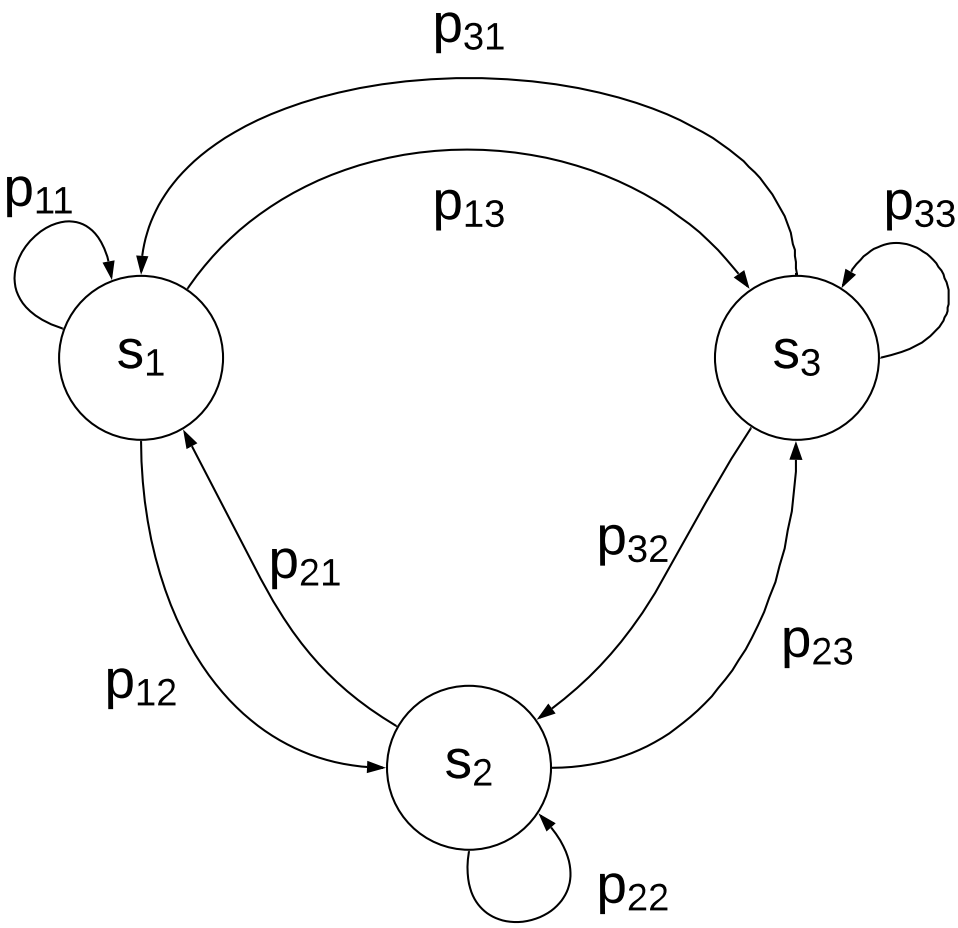
$$P = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1j} & \cdots \\ p_{11} & p_{12} & \cdots & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ p_{i1} & p_{i2} & \cdots & p_{ij} & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \end{pmatrix}. \tag{2.1}$$

## 2.3 Comportamento Baseado em Recompensas

A essência da AR é aprender através de interação, um agente de AR interage com o ambiente e, observando as consequências das suas ações, pode alterar seu próprio comportamento em respostas às recompensas recebidas. Esse paradigma de aprendizagem por tentativa e erro é uma das fundações da AR (SUTTON; BARTO, 2018). Outra influência na AR é o controle ótimo, que emprestou formalismos matemáticos, principalmente a programação dinâmica, que sustentam o campo.

Na configuração AR, um agente autônomo, controlado por um algoritmo de aprendizagem de máquina, observa um estado $s_t$ do ambiente no instante $t$. O agente interage com o ambiente realizando uma ação $a_t$ no estado $s_t$. Quando o agente faz uma ação, o ambiente e o agente fazem uma transição para um novo estado $s_{t+1}$, baseado no estado atual e na ação escolhida. O estado possui toda informação necessária para o agente tomar a melhor ação, que pode incluir partes do agente como posição e sensores. Na literatura de controle ótimo, estados e ações são geralmente representados por $\mathbf{x}_t$, e $\mathbf{u}_t$, respectivamente.

Figura 2.1 – Diagrama da cadeia de Markov

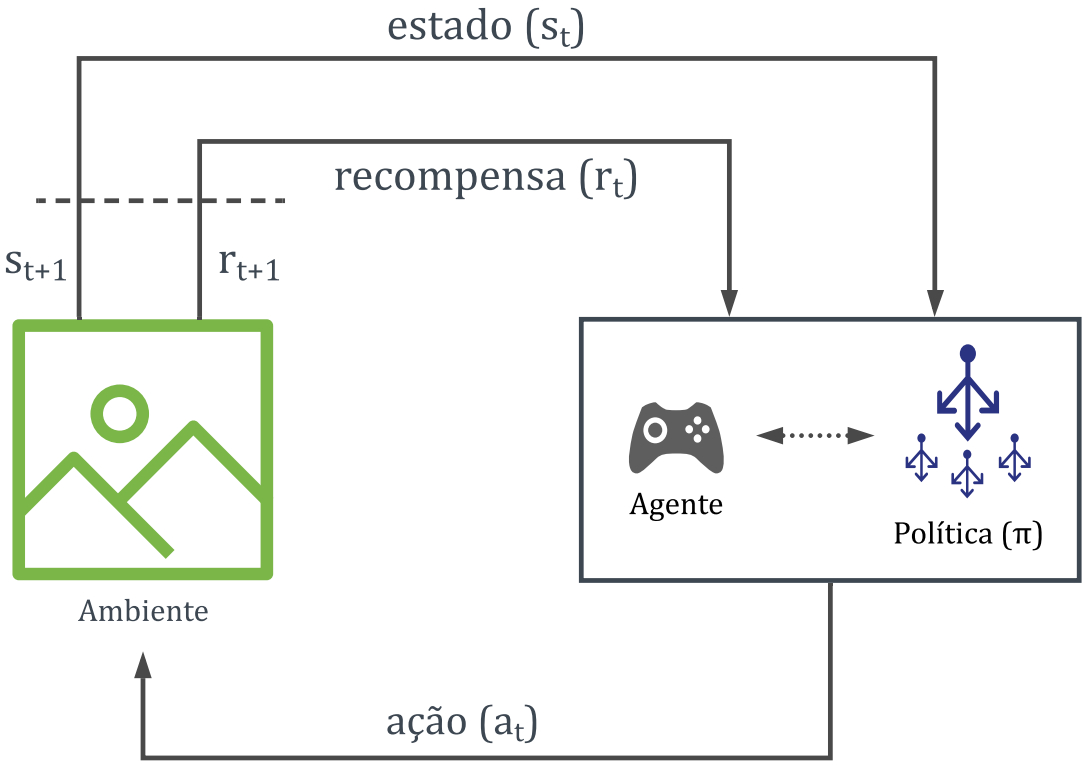



Fonte: Produzido pelo autor

A melhor sequência de ações é determinada pelas recompensas fornecidas pelo ambiente. Toda vez que o ambiente faz a transição para um novo estado, também fornece uma valor escalar de recompensa $r_{t+1}$ para o agente como um parecer. O objetivo do agente é aprender uma política (estratégia de controle) $\pi$ que maximize o retorno esperado (recompensa acumulada e descontada).

Dado um estado, a política fornece uma ação a ser tomada; uma política ótima é qualquer política que maximize o retorno esperado no ambiente. Nesse sentido, a AR visa resolver o mesmo problema como um controle ótimo. Entretanto, o desafio na AR é que o agente precisa aprender sobre as consequências das suas ações no ambiente por tentativa e erro, uma vez que diferente do controle ótimo, um modelo de transição de estados dinâmico não está disponível para o agente. Cada interação com o ambiente gera informação, que o agente usa para atualizar seu conhecimento. Esse processo é representado na Figura 2.2.

Figura 2.2 – Processo de percepção do agente




Fonte: Produzido pelo autor

## 2.4 Processo de Decisão Markoviano

Formalmente, AR pode ser descrita como um processo de decisão de Markov (PDM) (ALAGOZ et al., 2010), que consiste em:

- um conjunto de estados $S$ e uma distribuição de estados iniciais $p(s_0)$;
- um conjunto de ações $A$;
- uma função de recompensa imediata $R(s_t, a_t, s_{t+1})$;
- dinâmica de transição $\mathcal{T}(s_{t+1}|s_t, a_t)$ que mapeia o par estado-ação no instante $t$ em uma distribuição de estados no instante $t + 1$;
- um fator de desconto $\gamma \in [0, 1]$, onde valores mais próximos de zero priorizam recompensas imediadas.

No geral, a política $\pi$ é um mapeamento dos estados para uma distribuição probabilística sobre ações $\pi : S \rightarrow p(A = a|S)$. Se o PDM for episódico, ou seja, o estado é restaurado depois de cada episódio de período $T$, então a sequência de estados, ações, e recompensas num episódio constitui uma trajetória da política. Cada trajetória de uma política acumula recompensas do ambiente, resultando num retorno $R = \sum_{t=0}^{T-1} \gamma r_{t+1}$. O objetivo da AR é encontrar a política ótima, $\pi^*$ que alcança o retorno máximo esperado de todos os estados:

$$\pi^* = \arg\max_{\pi} E[R|\pi]. \tag{2.2}$$

Também é possível considerar PDMs não episódicos, onde $T = \infty$. Nessa situação $\gamma < 1$ previne o acúmulo de uma soma infinita de recompensas. Portanto, métodos que precisam de trajetórias completas não são aplicáveis, mas para aqueles que usam um conjunto finito e transições ainda são.

Um conceito importante sobre AR é a propriedade Markoviana, apenas o estado atual afeta o próximo estado, ou, em outras palavras, o futuro é condicionalmente independente do passado dado um estado presente. Isso significa que qualquer decisão feita em $s_t$ pode ser baseada somente em $s_{t+1}$ ao invés de $\{s_0, s_1, ..., s_{t-1}\}$. Mesmo que essa suposição seja realizada pela maioria dos algoritmos AR, é de certa forma irrealista, já que precisa que os estados sejam completamente observáveis. Uma generalização dos PDMs são os processos parcialmente observáveis, no qual o agente recebe uma observação, onde a distribuição da observação é dependente do estado atual e da ação anterior (KAELBLING; LITTMAN; CASSANDRA, 1998). No contexto de controle e processamento de sinais, a observação pode ser descrita como um mapeamento do modelo de estado espaço que dependa do estado atual e da última ação aplicada.

Algoritmos PDM parcialmente observáveis tipicamente mantém uma crença sobre o estado atual dado a crença do estado anterior, a ação realizada, e a observação atual.

## 2.5 Algoritmos AR

Existem basicamente três abordagens em resolver problemas com AR: métodos baseados em funções de valor, métodos baseados em busca de políticas e métodos baseados em modelos. Existem também abordagens híbridas que empregam uma combinação entre dois tipos de métodos. O algoritmo de AR desenvolvido nesse trabalho utiliza função de valor.

### 2.5.1 Funções de Valor

Métodos de funções de valor são baseados em estimar o valor (retorno esperado) de estar num dado estado. A função estado-valor $V^{\pi}(s)$ é o retorno esperado de iniciar no estado $s$ e seguir a política $\pi$:

$$V^{\pi}(s) = [R|s, \pi]. \tag{2.3}$$

A política ótima, $\pi^*$, tem uma função de estado valor $V^*(s)$ corespondente; a função estado-valor ótima pode ser definida como:

$$V^*(s) = \max_{\pi} V^{\pi}(s). \tag{2.4}$$

Se $V^*(s)$ está disponível, a política ótima pode ser obtida escolhendo a partir de todas as ações possíveis em $s_t$ e escolhendo a ação $a$ que maximize $E_{s_{t+1} \sim \mathcal{T}(s_{t+1}|s_t, a_t)}$. Na configuração AR, a dinâmica de transição $\mathcal{T}$ não é disponível. Portanto, nos construímos outra função, a função de valor estado-ação ou função de qualidade $Q^{\pi}(s, a)$, que é similar à $V^{\pi}$, com a diferença que a ação inicial $a$ é fornecida e a política $\pi$ só é seguida nos estados futuros:

$$Q^{\pi}(s, a) = [R|s, a, \pi]. \tag{2.5}$$

A melhor política, dado $Q^{\pi}(s, a)$, pode ser encontrada escolhendo $a$ de uma forma gulosa em cada estado: $\arg\max_a Q^{\pi}(s, a)$. Nessa política, pode-se definir $V^{\pi}(s)$ maximizando $Q^{\pi}(s, a)$:

$$V^{\pi}(s) = \max_a Q^{\pi}(s, a) = [R|s, a, \pi]. \tag{2.6}$$

Para encontrar $Q^{\pi}$, pode-se explorar a propriedade de Markov e definir a função conhecida como equação de Bellman (RUSSELL; NORVIG, 2009), que possui a seguinte forma recursiva:

$$Q^{\pi}(s_t, a_t) = E_{s_{t+1}}[r_{t+1} + \gamma Q^{\pi}(s_{t+1}, \pi(s_{t+1}))]. \tag{2.7}$$

Isso significa que $Q^{\pi}$ pode ser melhorada usando os valores atuais da estimativa de $Q^{\pi}$ para melhorar a própria estimativa. Isso é a base do *Q-learning* e do algoritmo *state-action-reward-state-action* (SARSA) (RUSSELL; NORVIG, 2009):

$$Q^{\pi}(s_t, a_t) \leftarrow Q^{\pi}(s_t, a_t) + \alpha\delta, \tag{2.8}$$

onde $\alpha$ é a taxa de aprendizagem e $\delta = Y - Q^{\pi}(s_t, a_t)$ erro de diferença temporal, $Y$ é o alvo como num problema de regressão. SARSA, um algoritmo *on-policy*, é usado para melhorar a estimativa de $Q^{\pi}$ usando transições geradas pelo comportamento da política, que resulta em $Y = r_t + \gamma Q^{\pi}(s_{t+1}, a_{t+1})$. *Q-learning* é *off-policy* uma vez que $Q^{\pi}$ é atualizada pelas transições que não necessariamentes foram geradas pela política derivada. Ao invés disso, *Q-learning* usa $Y = r_t + \gamma \max_a Q^{\pi}(s_{t+1}, a)$, que diretamente se apoxima do valor ótimo $Q^*$.

Para encontrar $Q^*$ de uma $Q^{\pi}$ arbitrária, pode-se usar iterações gerais de políticas, onde uma iteração de política consiste em uma avaliação da política e um melhoramento. Avaliação da política melhora a estimativa da função de valor, que pode ser alcançada minimizando os erros de diferença temporais das trajetórias percorridas pela política. Com a melhora da estimativa, a política pode naturalmente ser melhorada escolhendo ações de forma gulosa baseando na função de valor atualizada. Iteração geral de política permite intercalação dos passos para que o progresso possa ser feito mais rapidamente.

### 2.5.2 Busca de Política

Métodos de busca não precisam manter o modelo da função de valor mas diretamente procurar por uma política ótima $\pi^*$. Tipicamente, uma política parametrizada é escolhida, cujos parâmentros são atualizados para maximizar o retorno esperado $E[R|\theta]$ usando otimização baseada em gradiente ou sem gradiente (RUSSELL; NORVIG, 2009). Otimização sem gradiente pode cobrir espaços de parâmetros de baixa dimensão de forma eficiente, mas, treinamento baseado em gradiente continua como o método usado em algoritmos de AR com técnicas de aprendizagem profundo, sendo mais eficiente em termos de amostras quando políticas possuem um número grande de parâmetros.

Quando construindo a política diretamente, é comum produzir parâmetros para uma distribuição de probabilidade; para ações continuas, isso pode significar a média e a variância de distribuições Gaussianas, enquanto para ações discretas isso pode significar as probabilidades individuais de uma distribuição multinomial. O resultado é uma política estocástica da qual pode-se amostrar diretamente as ações. Com métodos sem gradiente, é necessário uma heurística para buscar numa classe de modelos predefinidos para encontrar políticas melhores. Métodos como estratégia evolutiva realizam *hill climbing*, num subespaço de políticas, enquanto que métodos mais complexos, como de busca em redes comprimidas, impõe tendências indutivas adicionais (RUSSELL; NORVIG, 2009).

### 2.5.3 Baseado em Modelo

Métodos baseados em modelos têm sua aprendizagem realizada indiretamente com um modelo do ambiente, realizando ações e observando os resultados dessas ações nesse

modelo de ambiente, assim o método utiliza um modelo preditivo do ambiente no seu processo de aprendizagem.

Alguns exemplos desses tipos de métodos são: Regulador Iterativo Linear Quadrático (iLQR), Modelo Preditivo de Controle (MPC) e Árvore de Busca de Monte Carlo (MCTS).

## 2.6 Algoritmos de Escalonamento

Algoritmos de escalonamento em sistemas de comunicação são utilizados para otimizar a utilização e compartilhamento dos recursos disponíveis (SADIQ; MADAN; SAMPATH, 2009). Alguns algoritmos de escalonamento conhecidos na literatura foram escolhidos para fazer uma comparação de desempenho ao longo desse trabalho, os quais pode-se citar: *Earliest Deadline First* (EDF), *Log rule* (LOG Rule), *Exponential rule* (EXP Rule) e seleção aleatória (SA) (SADIQ; MADAN; SAMPATH, 2009).

### 2.6.1 Earliest Deadline First

O algoritmo EDF prioriza processos que possuem um prazo de entrega mais curtos e, no caso de processos com mesmo prazo, processos que foram enfilerados primeiros tem prioridade. Algoritmos deste tipo possuem uma alta eficiência em sistemas mais simples, garantindo que todos os prazos sejam cumpridos. O uso deste algoritmo no sistema desse trabalho se resume a priorizar a alocação de recursos para dispositivos que possuam pacotes que chegaram há mais tempo.

### 2.6.2 EXP rule e LOG rule

A *EXP rule* estima o tempo de espera $w_k(i)$ para o dispositivo $k$ num determinado intervalo de tempo e cria uma fila de espera baseado nesses tempos de espera, a seleção do dispositivo ocorre de acordo com uma maximização do argumento, dado pela seguinte equação:

$$k^*(i) \in \arg\max_{1 \le i \le K} b_k \, exp\left(\frac{a_k w_k(i)}{1 + \sqrt{(1/K)\sum_j w_j(i)}}\right) \times SE_k(i), \tag{2.9}$$

onde $SE_k(i)$ representa a eficiência espectral do dispositivo $k$ e os parâmetros $a_k$, $b_k$ e $c$ são constantes positivas arbitrárias, ver (SHAKKOTTAI; STOLYAR, 2001) sobre como esses parâmetros devem ser escolhidos. O algoritmo faz uma seleção com base nos canais disponíveis sobre quais dispositivos entram na fila para execução do processo e quais ficam na fila de espera, a serem executados nos próximos períodos. Similarmente à regra

exponencial, o escalonador *LOG rule* é definida pela seguinte equação:

$$k^*(i) \in \arg\max_{1 \leq k \leq K} b_k \, log\,(c + a_k w_k(i)) \times SE_k(i), \tag{2.10}$$

os parâmetros $a_k$, $b_k$ e $c$ e os valores de $SE_k(i)$ são similares aos da equação da *EXP rule* e maiores informações sobre eles também podem ser encontradas em (SHAKKOTTAI; STOLYAR, 2001). De maneira similar ao algoritmo *EXP rule*, a *LOG rule* determina a fila de execução, escolhendo quais dispositivos são executados no instante atual e quais entram na fila de espera, com a diferença de utilizar uma equação que usa uma expressão logarítmica para o cálculo dos tempos de prioridade.

### 2.6.3 Seleção Aleatória

O agente da seleção aleatória realiza o escalonamento de pacotes no sistema IdC de forma aleatória, independente do estado que o sistema se encontra.

## 2.7 Trabalhos Relacionados

Pode-se encontrar na literatura trabalhos relevantes e similares, como por exemplo em (NAPARSTEK; COHEN, 2018), onde múltiplos usuários dividem múltiplos canais ortogonais em um sistema de comunicação sem fio e se deseja maximizar a utilidade da rede multiusuário do sistema enquanto lida com um conjunto grande de estados, o trabalho é similar mas se diferencia por focar na colisão de usuários, ou seja, em evitar que mais de um usuário tente transmitir por um mesmo canal, o algoritmo é treinado calculando os canais possíveis de transmissão para cada usuário sequencialmente em um mesmo instante de tempo.

No trabalho (ZHANG; XU, 2019), pode-se observar o uso de aprendizagem por reforço profundo num sistema NOMA, entretanto o trabalho aborda um sistema onde as subportadoras consigam suportar mais de um usuário, assim, todos os usuários podem transmitir e as ações se aplicam a todos os usuários conjuntamente, diferente da proposta do presente trabalho.

O artigo (Wei et al., 2020) tem uma proposta de uma utilização de aprendizagem por reforço ampla, aplicada num cenário internet das coisas autônoma rápida, mais especificamente numa utilização de controle inteligente de semáforos de uma cidade inteligênte para o controle de trânsito de veículos e faz uma comparação entre os tempos de espera do tráfego dos uso fixo dos sinais de trânsito e algoritmos com aprendizagem por reforço ampla e profunda.

Em (KHURPADE; RAO; SANGHAVI, 2018), pode-se encontrar um estudo sobre os requisitos para a internet das coisas, as vantagens e desvantagens do uso do 5G. O

trabalho mostra como o 5G pode ser um suporte para os cenários IdC.

O trabalho (HASEGAWA et al., 2020) mostra uma avaliação comparativa experimental num cenário real de IdC com dois sistemas coexistindo no mesmo local, os sistemas do experimento utilizam aprendizagem de máquina para a seleção de canal dos dispositivos. O artigo conclui que o uso de aprendizagem por reforço possui os melhores resultados de sucesso de entrega de pacotes e taxa de confiabilidade.

Em (XIONG et al., 2020) pode-se observar um uso de aprendizagem por reforço para determinar alocação de recursos em um cenário IdC em computação de borda e uma proposta nova de *Q-network* que obteve menores valores de perda e maiores valores médios de recompensas, resultando numa melhor performance.

# 3 Metodologia

Neste capítulo são abordados os modelos utilizados para a construção dos estados do sistema, o equacionamento dos valores de utilidade do sistema, necessários para o treinamento do agente AR e o algoritmo AR proposto utilizando *Q-learning*.

## 3.1 Modelagem do Sistema

O sistema IdC cognitivo mostrado na Figura 3.1 consiste em uma estação base com $M$ canais de transmissão disponíveis para $K$ dispositivos que transmitem pacotes de dados. O tempo é discretizado em intervalos iguais e em cada intervalo de tempo, pacotes chegam à cada dispositivo que os transmite caso haja um canal disponível. O sistema é capaz de transmitir pacotes de mais de um dispositivo ao mesmo tempo. Cada dispositivo possui um *buffer* de tamanho $L$ para armazenar pacotes que não foram transmitidos, onde os pacotes chegam obedecendo uma distribuição de Poisson com a taxa de chegada $\lambda$ e são transmitidos por um dos $M$ canais com uma taxa de codificação $V$, não há transmissão quando a qualidade do canal é mínima. Quando pacotes chegam ao *buffer* de um dispositivo que está cheio, há perda de pacotes.

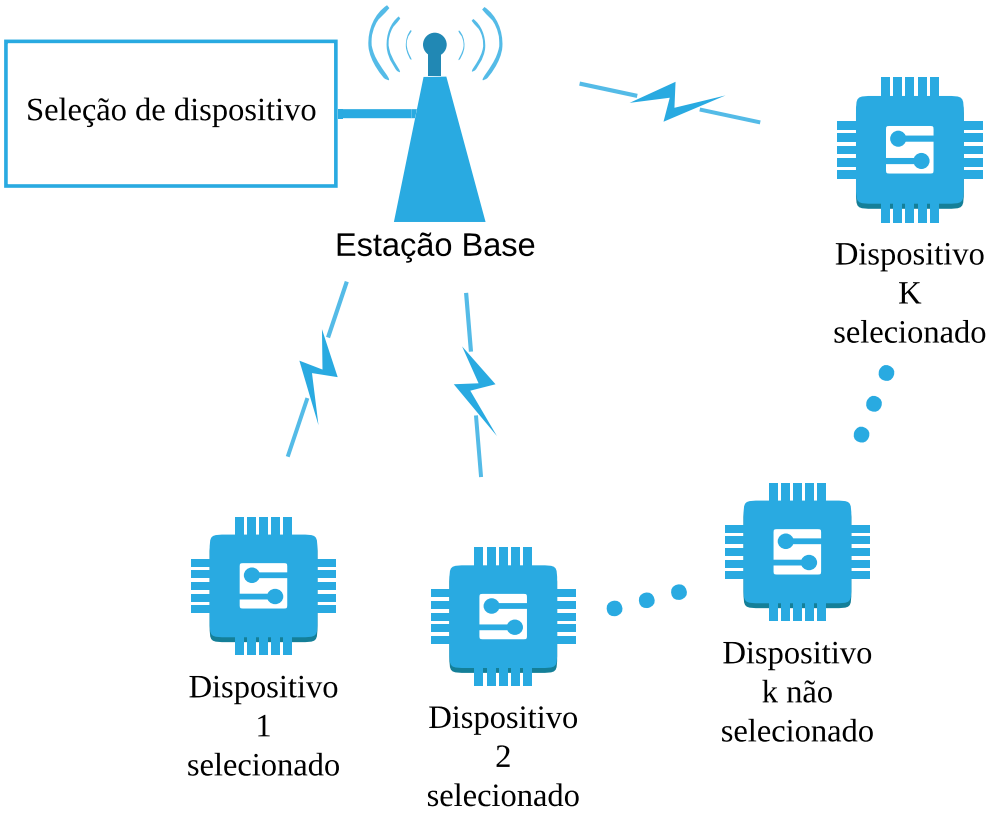
O sistema é considerado cognitivo pelo fato da estação base possuir informações sobre o estado dos dispositivos conectados à ela durante todos os instantes de tempo, e é capaz de tomar decisões e executar ações com base nesses dados (VERNON, 2014). Os dispositivos IdCs considerados são do tipo que recebem dados e transmitem à estação base, podendo ser sensores ou dispositivos de monitoramento, ou mesmo *gateways* que recebem os dados de outros dispositivos e os transmitem para a estação base.

## 3.2 Estado dos *Buffers*

Cada um dos $K$ dispositivos possuem um *buffer* de tamanho $L$, e em cada intervalo do sistema pacotes chegam e são transmitidos pelo dispositivo. Assim, o *buffer* de cada dispositivo pode apresentar filas de pacotes que variam de tamanho entre 0 e $L$ pacotes. A probabilidade de chegar $d$ pacotes em um determinado intervalo de tempo $i$ é de $p(d_i) = exp(-\lambda)\lambda^{d_i}/d_i!$, onde $\lambda$ é a taxa de chegada em pacotes por intervalo de tempo. A quantidade de pacotes no *buffer* no intervalo de tempo posterior pode ser dada por:

$$l_{i+1,k} = min(d_{i,k} + l_{i,k} - t_{i,k}, L), \tag{3.1}$$

Figura 3.1 – Esquemático do funcionamento do sistema




Fonte: Produzido pelo autor

onde $i$ é o intervalo de tempo, $l_{i,k}$ é a quantidade de pacotes para o dispositivo $k$, $d_{i,k}$ é o número de pacotes que chegam e $t_{i,k}$ a quantidade de pacotes que são transmitidos.

A probabilidade de transição dos estados dos *buffers* dos $K$ dispositivos é o produtório das probabilidades individuais de cada dispositivo, ou seja:

$$p_l(l, l') = \prod_{k=1}^{K} p_{l,k}(l_i, l_{i+1}|a_{i,k}), \tag{3.2}$$

onde $a_{i,k}$ é o número de pacotes transmitidos pelo dispositivo $k$ no intervalo $i$.

## 3.3 Estado dos Canais

Assumindo que a relação sinal-ruído (SNR) obedece a distribuição de Rayleigh (ZAIDI et al., 2018), cuja função de densidade de probabilidade é $p(\rho) = 1/\bar{\rho}\ exp(-\rho/\bar{\rho})$, com o parâmetro $\rho > 0$ e $\bar{\rho} = E(\rho)$ sendo a SNR média. Seja o limiar da SNR expressado

como $\rho_{snr} = \{\rho_1, \rho_2, \ldots, \rho_{C-1}\}$ e $C$ o número de estados dos canais, pode-se obter a probabilidade de distribuição do estado do canal como:

$$p_C(c_n) = \int_{\rho_n}^{\rho_{n+1}} \mathrm{p}(\rho)\, \mathrm{d}\rho. \tag{3.3}$$

Assim, a probabilidade de transição do estado do canal é (GREWAL; KRZYWINSKI; ALTMAN, 2019):

$$p_C(c_n, c_{n+1}) = N(\rho_{n+1}) T_f / p_C(c_n), \tag{3.4}$$

onde $n \in \{1, 2, \ldots, N-2\}$, e:

$$p_C(c_n, c_{n-1}) = N(\rho_n) T_f / p_C(c_n), \tag{3.5}$$

onde $n \in \{1, 2, \ldots, N-1\}$ e $N(\rho_n) = \sqrt{2\pi\rho_n/\bar{\rho}} f_d$ com $f_d$ sendo o efeito doppler máximo. Analogamente à equação 3.2, a probabilidade de transição dos $M$ canais é

$$p_C(c, c') = \prod_{m=1}^{M} p_{c,m}(c_i, c_{i+1}). \tag{3.6}$$

## 3.4 Potência

A transmissão de pacotes é realizada por diferentes modos de transmissão $j \in \{0, 1, ..., J\}$, onde os modos 0 e 1 representam nenhuma transmissão e transmissão BPSK respectivamente, e para $j \geq 2$, $2^j-$QAM. Pode-se então estimar a potência mínima de transmissão $P$ no estado de canal $c_i$ com a modulação $j$ a partir da taxa de erros de bit $p_{BER}$ (HAYKIN, 2013):

$$p_{BER}(c_i, j) \leq 0,5 erfc(\sqrt{\rho_i P(c_i, j)/WN_0}), \tag{3.7}$$

para $j = 1$, e para $j \geq 2$, tem-se:

$$p_{BER}(c_i, j) \leq 0,2 exp(-1,6\rho_i P(c_i, j)/WN_0(2^j - 1)), \tag{3.8}$$

onde $WN_O$ é a potência de ruído.

## 3.5 O Sistema Como uma Cadeia de Markov

O sistema descrito na seção anterior, pode ser modelado como uma cadeia de Markov, uma vez que o estado seguinte depende somente do estado atual e da ação escolhida pelo agente (GREWAL; KRZYWINSKI; ALTMAN, 2019). Como o sistema permite que a transmissão ocorra de modo simultâneo, durante cada intervalo de tempo, o

agente deverá escolher até no máximo $M$ dispositivos para fazer a transmissão de seus respectivos pacotes, através dos $M$ canais disponíveis, assumindo $M < K$, e usando modulações diferentes para cada canal. Assim, o número total de ações possíveis em um determinado estado é o produto de duas permutações sem repetições:

$$A = \frac{(J+1)!}{(J+1-M)!}\frac{K!}{(K-M)!}, \tag{3.9}$$

onde $J$ é o número máximo dos modos de transmissão $2^j-$QAM do sistema.

## 3.6 Probabilidade de Transição de Estados

Os estados do sistema são definidos por uma combinação dos estados dos *buffers* de cada dispositivo com os estados dos canais do sistema. Ambos podem ser definidos como uma permutação com repetição, uma dos $K$ dispositivos tomados dos estados possíveis para cada dispositivo, e outra, dos $M$ canais tomados dos estados possíveis dos canais. Assim, o número total de estados do sistema é:

$$S = (L+1)^K C^M, \tag{3.10}$$

onde $L$ é o tamanho máximo do *buffer* e C é o número de estados dos canais. Assim, a probabilidade de transição de estados do sistema é:

$$p_S(S_i, S_{i+1}|a_i) = \prod_{k=1}^{K} p_{l_k}(l_i, l_{i+1}|a_i) \times \prod_{m=1}^{M} p_S(c_i, c_{i+1}). \tag{3.11}$$

## 3.7 Utilidade do Sistema

Para a escolha da recompensa do treinamento, uma expressão que representa a utilidade do sistema foi criada com a finalidade de avaliar três características de interesse: a vazão de pacotes de dados, a potência consumida pelo sistema, e o uso dos *buffers*

Em cada intervalo de tempo $i$, a vazão é definida como sendo o somatório do produto entre a taxa de codificação $V$ e o modo de transmissão $j$ para cada um dos $K$ dispositivos, e o custo é definido como o produto da potência de transmissão consumida $P_{s_i}(s_i, a_i)$ com o somatório do valor da pressão do *buffer* $f_{i,k} = exp(\theta \times l_{i,k})$, com $\theta$ sendo o coeficiente de pressão.

A utilidade do sistema é diretamente proporcional ao número de pacotes transmitidos e inversamente proporcional à pressão dos *buffers* e do consumo de potência, sendo

representada pela seguinte equação:

$$O(s_i, a_i) = \frac{\sum_{k=1}^{min(K,M)} V \times j_k}{\left(\sum_{k=1}^{K} f_{i,k}\right) P_{s_i}(s_i, a_i)}. \tag{3.12}$$

## 3.8 Aprendizagem por Reforço

A aprendizagem 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 (XIONG et al., 2019). 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 trabalho, o algoritmo de aprendizagem por reforço *Q-learning* é utilizado, no qual é necessário obter as probabilidades de transição de estados e as recompensas de cada ação possível. Deve-se selecionar um fator de desconto que indica a relevância das recompensas futuras. Uma matriz $\mathbf{Q}$ de ação-valor é então gerada, com valores de utilidade esperados para cada ação realizada em cada estado, a matriz é atualizada a medida em que novas políticas são testadas. Para cada política $\pi$ existe um valor agregado $V^{\pi}(s_i)$ e o objetivo do treinamento é fazer o agente aprender a determinar uma política que maximize $V^{\pi}(s_i)$ (XIONG et al., 2019).

O seguinte algoritmo foi utilizado para treinar o agente com aprendizagem por

reforço:

**Algoritmo 1:** Algoritmo *Q-learning* baseado em Modelagem Markoviana do Sistema

Calcule os valores da matriz $\mathbf{P}$ de probabilidade de transição de estados de acordo com a equação 3.11.

Inicialize os valores da matriz $\mathbf{R}$ de recompensas usando a utilidade do sistema usando a equação 3.12.

Inicialize a matriz $\mathbf{Q} = \mathbf{0}$.

**for** $j = 1 : N$ **do**

  Selecione um estado incial $s_0$ aleatório

  **for** $i = 1 : N$ **do**

    Selecione uma ação aleatória $a_i$ que seja possível de ser realizada no estado atual $s_i$ e que leva o sistema ao novo estado $s_{i+1}$.

    Atualize o valor de $\mathbf{Q}$:

    $Q_{s_i,a_i} \leftarrow Q_{s_i,a_i} + \alpha(R_{s_i,a_i} + \gamma max(Q_{s_{i+1}}) - Q_{s_i,a_i})$,

    onde $\alpha$ é a taxa de aprendizagem do algoritmo e $\gamma$ é o fator de desconto.

  **end for** $i$

**end for** $j$

Diferentes políticas podem ser criadas utilizando propostas diferentes na inicialização dos valores das recompensas, assim, pode-se priorizar outros aspectos de um mesmo sistema. Nesse artigo, a utilidade do sistema é utilizada como recompensa imediata das ações no treinamento do agente.

Com a matriz $\mathbf{Q}$ devidamente atualizada com o treinamento, faz-se então, através de simulações, a comparação do desempenho da abordagem proposta com o desempenho de outros algoritmos de escalonamento conhecidos da literatura. Estes algoritmos podem ser encontrados na sessão 2.6.

# 4 Simulações e Resultados

Neste capítulo são apresentados os parâmetros e as considerações feitas para a realização das simulações. Também são mostrados os resultados comparativos entre os algoritmos considerados.

## 4.1 Parâmetros da Simulação

Para verificar e comparar a eficiência do algoritmo proposto utilizando AR, foram realizadas simulações computacionais com outros algoritmos na literatura agindo sobre o mesmo sistema IdC cognitivo. O sistema assim como os cenários e algoritmos considerados foram implementados no Matlab 2019b.

Um problema encontrado na execução do algoritmo proposto foi a necessidade de armazenar os valores de probabilidade de transição de estados na matriz $\mathbf{P}$ o que limita os valores dos parâmetros das equações 3.9 e 3.10, pois sistemas com altos valores de estados e ações acabam gerando um arquivo para a matriz $\mathbf{P}$ que ultrapassava os limites de tamanho do *software* utilizado.

Nas simulações do sistema IdC foram considerados 4 cenários diferentes, o primeiro cenário tem o objetivo de observar o funcionamento do algoritmo quando mais pacotes de dados chegam aos dispositivos, o segundo cenário mostra o comportamento quando mais dispositivos são adicionados à rede, no terceiro, pode-se observar o algoritmo agindo num sistema com recursos limitados. E por fim, o quarto cenário consiste no funcionamento do sistema quando se melhora a qualidade do canal aumentando o valor do parâmetro $\rho$.

Os parâmetros comuns aos quatro cenários simulados na simulação podem ser encontrados na Tabela 4.1, os demais parâmetros necessários para simulação podem ser encontrados nas subseções específicas de cada cenário.

Tabela 4.1 – Parâmetros usados na simulação.

| Parâmetros | Valor |
|---|---|
| Coeficiente de pressão do *buffer* | $\theta = 0,5$ |
| Limite de BER | $BER \leq 10^{-3}$ |
| Potência de ruído | $10^{-3}WN_0/W$ |
| Número total de intervalos de tempo | 100 |
| Frequência do efeito Doppler | $f_d = 50Hz$ |
| Coeficiente de desconto | $\gamma = 0,5$ |
| Taxa de aprendizagem Q | $\alpha = 1/\sqrt{n_{ite} + 2}$ |
| Iterações do treinamento | $N = 100000$ |

As Figuras 4.1 à 4.24 apresentam gráficos comparativos relacionados ao desempenho dos 5 tipos de alocação: Seleção Aleatória (SA), *Earliest Deadline First* (EDF), Regras Logarítmica (*Log Rule*) e Exponencial (*EXP Rule*) e o algoritmo que utiliza Aprendizagem por Reforço (AR) e Modelagem Markoviana.

### 4.1.1 Cenário 1 - Taxa de Chegada

O primeiro cenário consiste em observar o comportamento dos algoritmos de alocação de recursos variando a taxa de chegada $\lambda$ de 0 a 1 pacotes/ms. O valor do intervalo de tempo de 1 milissegundo foi adotado por ser o intervalo de tempo de transmissão (TTI) em tecnologias 4G e corresponde ao tamanho de um *slot* na numerologia 0 em 5G (ZAIDI et al., 2018). Nesse cenário, o sistema possui $K = 4$ dispositivos com um *buffer* de tamanho $L = 3$, $M = 3$ canais disponíveis, com $C = 2$ estados de canal, $J = 4$ modos de transmissão, ou seja, as modulações possíveis para a transmissão são BPSK, 4-QAM, 8-QAM, 16-QAM e nenhuma transmissão. A taxa de codificação considerada é $V = 2$ e o parâmetro de Rayleigh é $\rho = 0{,}2$.

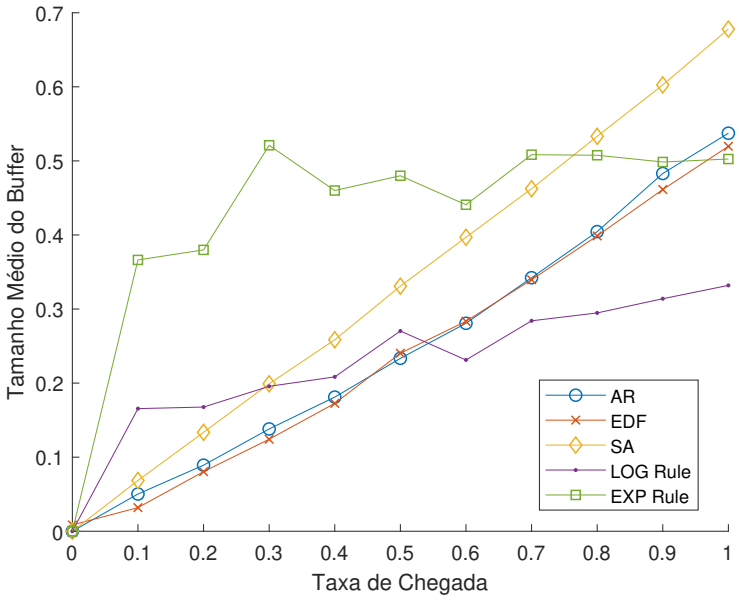
Os resultados do tamanho médio da fila no *buffer* em relação à taxa de chegada para os algoritmos considerados são apresentados na Figura 4.1. Nota-se que ao se aumentar a taxa de chegada, os algoritmos apresentam maiores valores de tamanho da fila no *buffer* devido à maior quantidade de pacotes que chegam e são armazenados na fila. Os menores valores de tamanho médio da fila foram apresentados pelos algoritmos EDF, *Log Rule* e pelo algoritmo proposto AR baseado em Cadeias de Markov.

Os resultados dos valores da utilidade do sistema para os algoritmos considerados nas simulações são apresentados na Figura 4.2. Verifica-se que com o aumento da taxa de chegada, os algoritmos apresentam menores valores de utilidade, ocasionadas pela necessidade de aumento da potência pelo aumento do número de pacotes transmitidos.

O algoritmo proposto apresentou os maiores valores de utilidade dados pela equação 3.12. Um algoritmo que apresente maior valor de utilidade representa que o mesmo controla a transmissão de pacotes de forma mais eficiente, com uma melhor relação vazão e consumo de potência dos dispositivos. Os três parâmetros que compõem a expressão da utilidade do sistema serão isolados e comparados também para um maior entendimento do comportamento dos algoritmos.

Na Figura 4.3 pode-se observar a potência média em $mW$ consumida pelo sistema ao se transmitir os pacotes de dados durante o intervalo de tempo simulado. O algoritmo de AR proposto possui os menores valores de consumo de potência, entretanto apenas com as informações da potência não se pode ter uma inferência completa do comportamento do algoritmo, por exemplo, a seleção aleatória é o algortimo com menores valores depois do proposto. É preciso conhecer a relação do consumo com a vazão de pacotes.

Figura 4.1 – Tamanho médio do *buffer* em relação à taxa de chegada para o cenário 1




Fonte: Produzido pelo autor

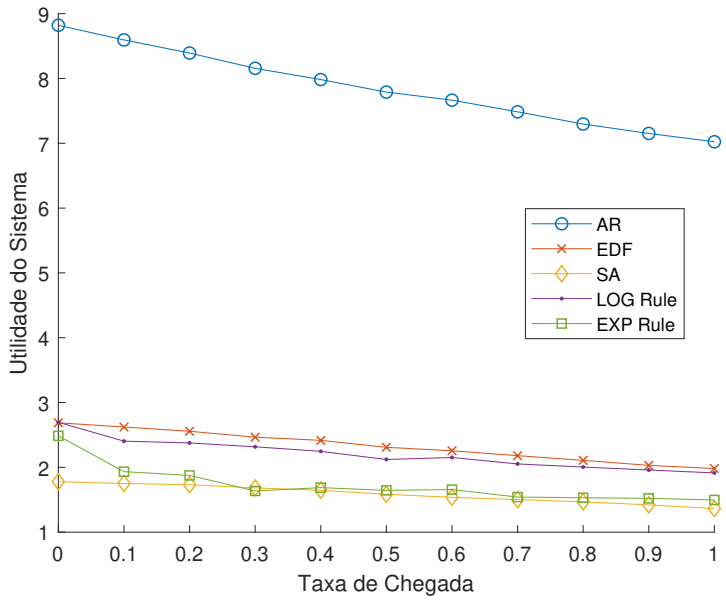
Pode-se ver os dados acerca da pressão dos *buffers* na Figura 4.4. A pressão de *buffer* é calculada usando o parâmetro $\theta$ (coeficiente de pressão) e é um dos três parâmetros que compõem a utilidade do sistema na equação 3.12 contribuindo negativamente para os valores da mesma. Os algoritmos AR e EDF possuem os menores valores para taxa de chegada de até 0,5 pacotes por segundo, e para $\lambda$ acima disso o *Log Rule* possui os menores valores.

Na Figura 4.5, visualizam-se os resultados da relação entre a taxa de pacotes perdidos pela taxa de pacotes transmitidos dos algoritmos considerados na simulação. Os algoritmos EDF e AR apresentaram os menores valores de taxas de perda de pacotes ao longo da variação da taxa de chegada.

Na Figura 4.6, pode-se observar os resultados normalizados para a quantidade de pacotes de dados transmitidos. Uma característica interessante acerca desse gráfico é a alta taxa de transmissão dos algoritmos *EXP* e *Log Rule* que não deveria refletir na alta taxa de pacotes perdidos dos mesmo algoritmos observados na Figura 4.5, isto acontece por conta do viés que esses algoritmos tem em privilegiar os primeiros dispositivos

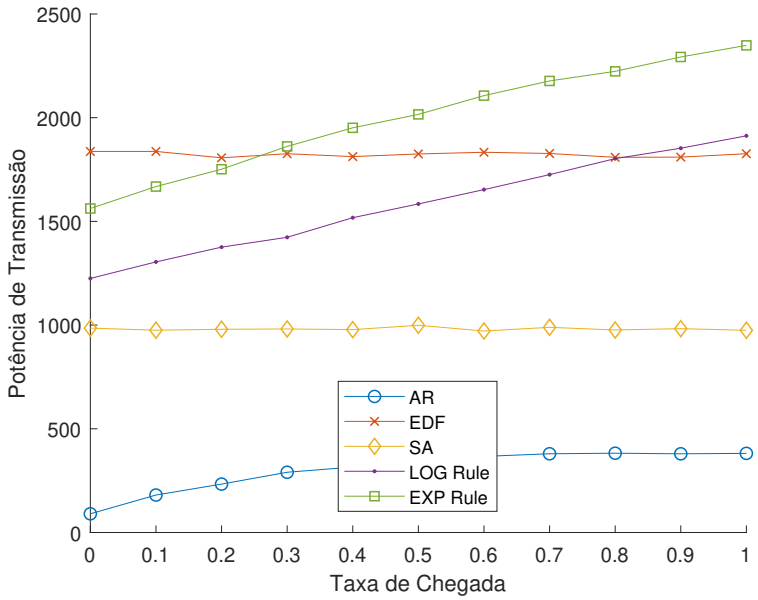
Com os dados do cenário 1, pode-se concluir que os valores de utilidade do algoritmo proposto ocorrem principalmente por conta da contribuição da potência consumida, que no caso do algoritmo AR são os menores dentre os algoritmos simulados. Os menores valores de potência obtidos pelo algoritmo AR não compromete o desempenho como a perda de pacotes.

Figura 4.2 – Utilidade do sistema em relação à taxa de chegada para o cenário 1



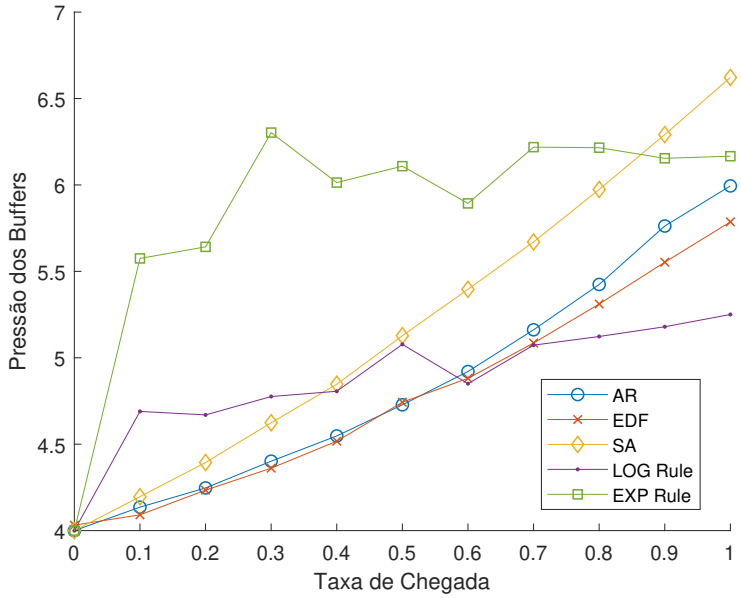

Fonte: Produzido pelo autor

Figura 4.3 – Potência consumida em $mW$ pelo sistema em relação à taxa de chegada para o cenário 1



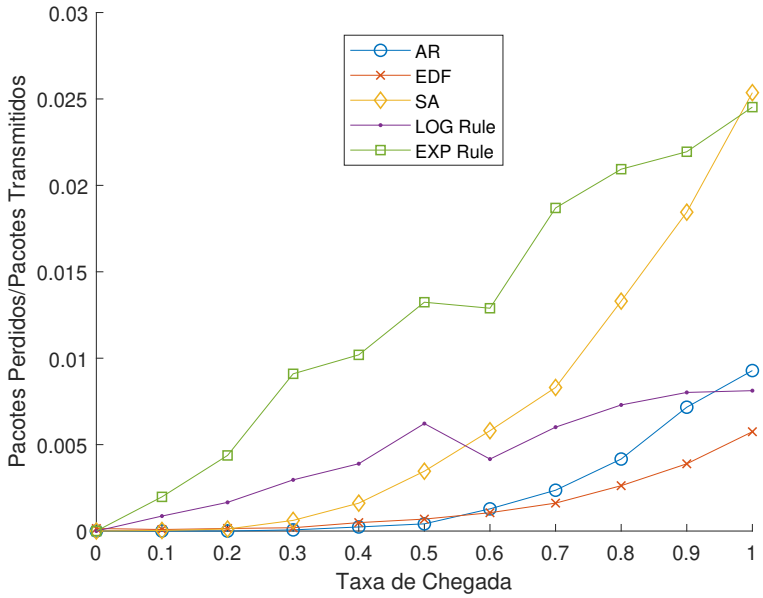

Fonte: Produzido pelo autor

Figura 4.4 – Pressão média dos *buffers* em relação à taxa de chegada para o cenário 1



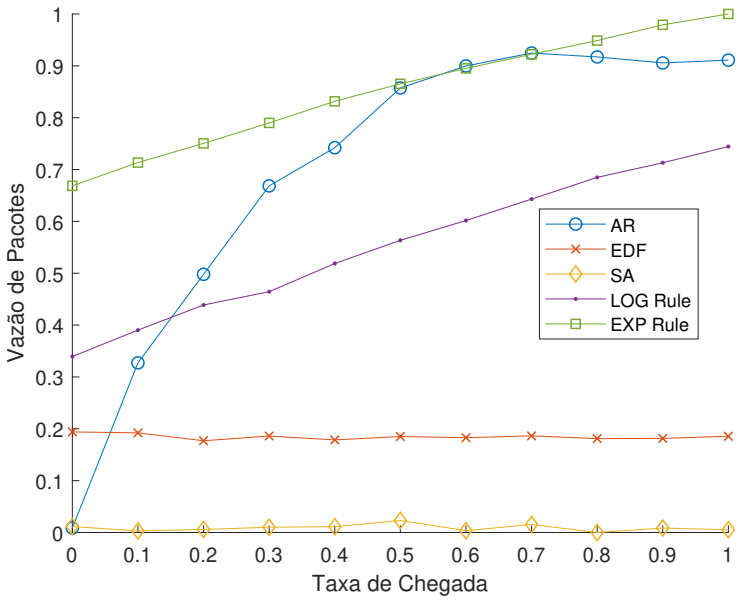

Fonte: Produzido pelo autor

Figura 4.5 – Taxa de pacotes perdidos por pacotes transmitidos em relação à taxa de chegada para o cenário 1




Fonte: Produzido pelo autor

Figura 4.6 – Quantidade de pacotes transmitidos em relação à taxa de chegada para o cenário 1




Fonte: Produzido pelo autor

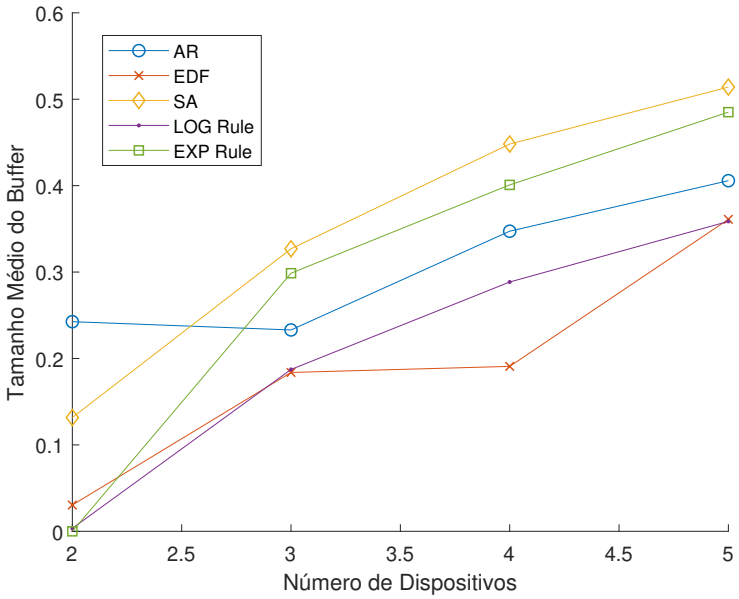
### 4.1.2 Cenário 2 - Dispositivos

Para o segundo cenário, fixa-se o parâmetro da taxa de chegada em $\lambda = 0{,}5$ pacotes a cada milissegundo, altera-se o número de canais disponíveis para $M = 2$ e realiza-se a variação da quantidade de dispositivos, com $K$ variando de 2 à 5 dispositivos, os demais parâmetros possuem os mesmos valores do cenário 1.

A maioria dos gráficos deste cenário possui um valor divergente no algoritmo proposto quando aplicado a uma situação de $K = 2$ e $M = 2$, pois o algoritmo foi criado com o número de canais disponíveis menor que o número de dispositivos. Entretanto se manteve esses dados para comparação com os outros algoritmos que não possuem tal condição de funcionamento.

O tamanho médio da fila do *buffer* é mostrado na Figura 4.7. Com o aumento do número de dispositivos no sistema, há um aumento na quantidade de pacotes na fila do *buffer* de todos os algoritmos, os menores valores foram dos algoritmos AR e EDF exceto para 2 dispositivos no qual o algoritmo proposto possui o maior valor de pressão média de *buffer*, pelo motivo citado anteriormente.

Figura 4.7 – Tamanho médio do *buffer* em relação ao número de dispositivos para o cenário 2

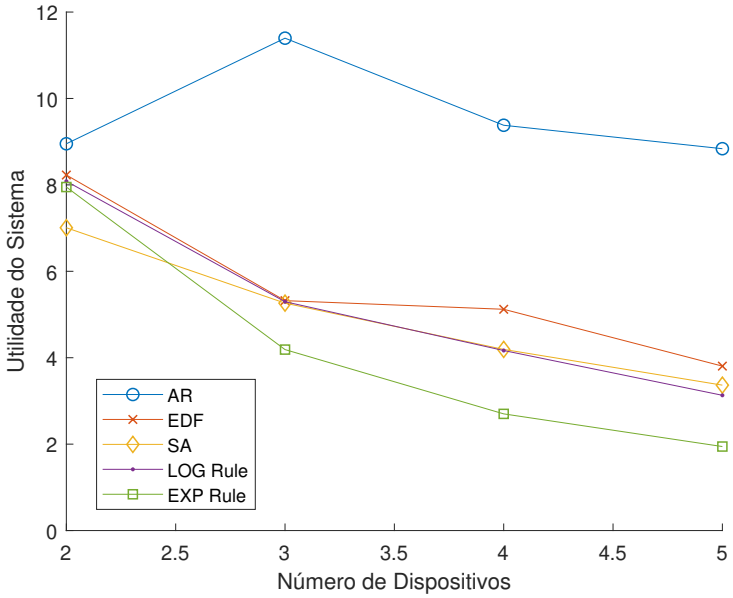



Fonte: Produzido pelo autor

A utilidade do sistema, mostrado na Figura 4.8, decresce com o aumento de dispositivos devido à dificuldade em se controlar a pressão dos *buffers*, que aumenta o custo do sistema. Pode-se observar a transição de 2 para 3 dispositivos no qual o algoritmo AR aumenta sua utilidade, o que também pode ser explicado de acordo com a não otimização

do algoritmo com $K = 2$.

Figura 4.8 – Utilidade do sistema em relação ao número de dispositivos para o cenário 2




Fonte: Produzido pelo autor

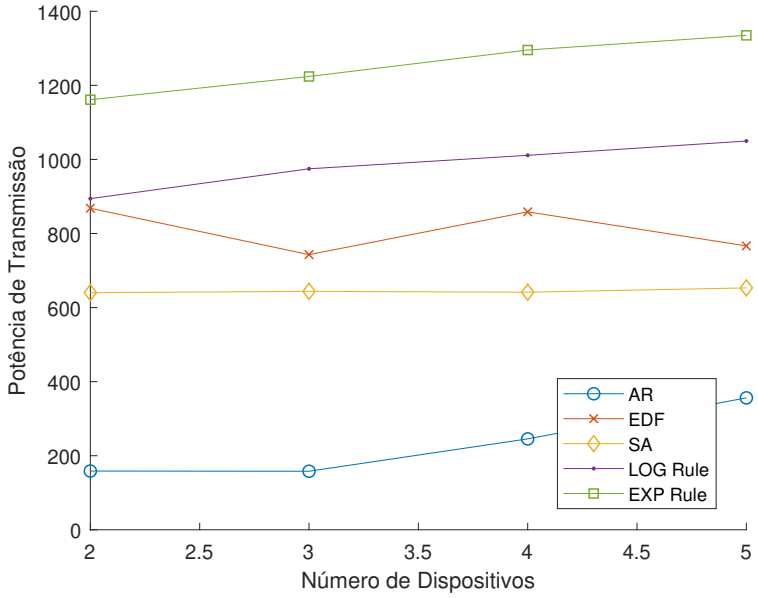
Assim como no cenário 1, a potência consumida é o fator que mais influencia na utilidade do sistema. Como se pode observar na Figura 4.9 possuindo valores muito menores que os outros algoritmos, incluindo o aleatório. Neste gráfico pode se ver o comportamento alternado do algoritmo EDF que possui esse comportamento por conta da sua própria lógica, como o algoritmo EDF cria uma fila de prioridade e essa fila só é atualizada quando ela é terminada, pode haver ociosidade quando o número de dispositivos não é múltiplo do número de canais disponíveis, isso faz com que para dispositivos ímpares (uma vez que $M = 2$) o sistema pode fazer ações ociosas no algoritmo EDF. O mesmo comportamento pode ser observado em outros gráficos.

Na Figura 4.10 pode se observar que os valores de pressão dos *buffers* de todos algoritmos são próximos entre si, com o EDF e o AR possuindo os menores valores.

As taxas de pacotes perdidos por pacotes transmitidos são mostrados na Figura 4.11, que crescem com o aumento do número de dispositivos. Os algoritmos que proporcionaram menos perdas de pacotes para o sistema foram o AR e o EDF.

Os valores normalizados da vazão de pacotes da Figura 4.12 mostra que há um aumento na vazão quando se aumenta o número de dispositivos. Pode-se observar um aumento pronunciado na vazão do algoritmo AR em relação aos demais. Neste gráfico também é possivel observar claramente o comportamento alternado do algoritmo EDF entre números pares e ímpares de dispositivos.

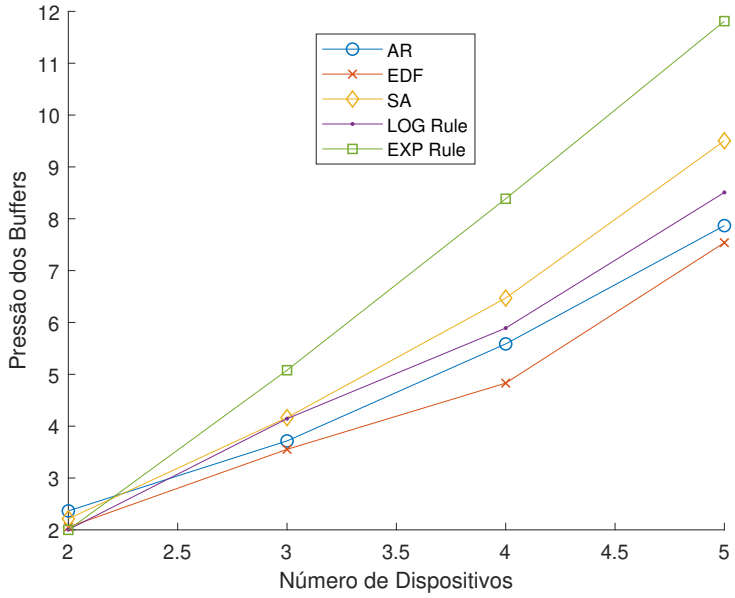
Figura 4.9 – Potência consumida em $mW$ pelo sistema em relação ao número de dispositivos para o cenário 2




Fonte: Produzido pelo autor

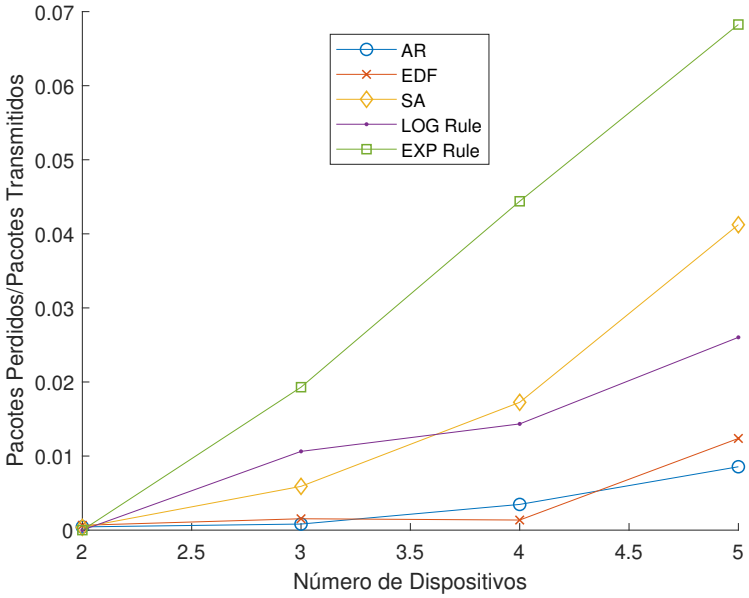
Fazendo uma análise dos dados do cenário 2, pode-se inferir que o algoritmo proposto consegue economizar na potência consumida e ainda manter uma baixa taxa de perda de pacotes. O algoritmo EDF também possui valores próximos de pacotes perdidos, entretanto por consumir maior potência, sua utilidade é inferior à do algortimo AR.

Figura 4.10 – Pressão média dos *buffers* em relação ao número de dispositivos para o cenário 2



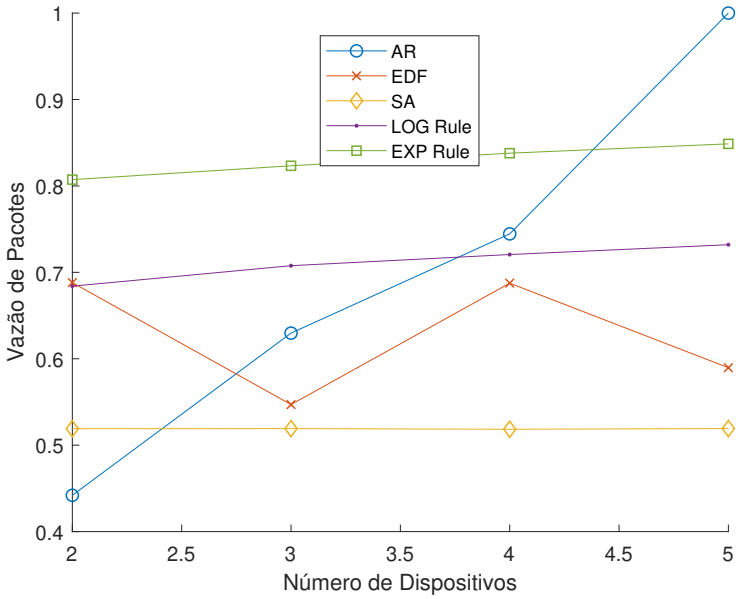

Fonte: Produzido pelo autor

Figura 4.11 – Taxa de pacotes perdidos por pacotes transmitidos em relação ao número de dispositivos para o cenário 2




Fonte: Produzido pelo autor

Figura 4.12 – Quantidade de pacotes transmitidos em relação ao número de dispositivos para o cenário 2




Fonte: Produzido pelo autor

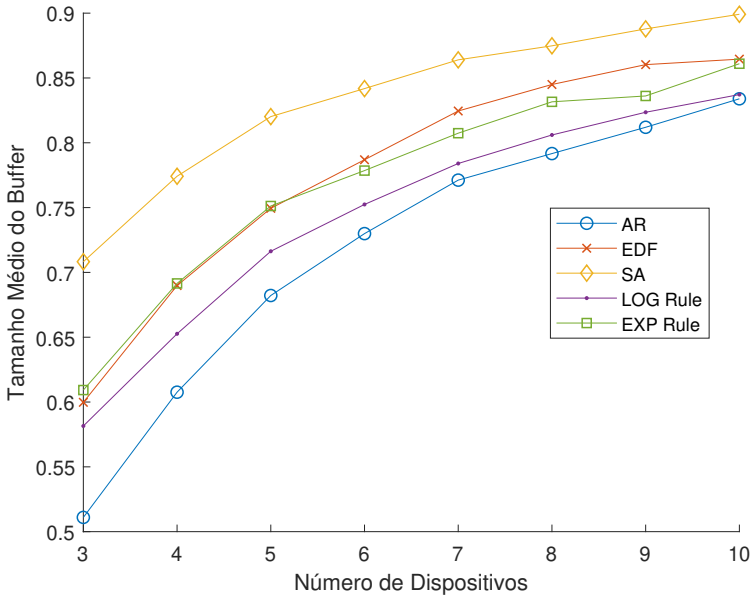
### 4.1.3 Cenário 3 - Dispositivos com Recursos Limitados

Para o terceiro cenário, aumenta-se o número de dispositivos variando entre $K = 3$ e $K = 10$, enquanto que se reduz o número de canais para $M = 1$ canal disponível, o tamanho de *buffer* de é reduzido para $L = 1$, e $J = 2$ modos de transmissão de pacotes, neste cenário, a simulação observa o comportamento com a taxa de chegada $\lambda$ de 1 pacote/ms.

Este cenário possui limitações nos seus parâmetros de número de canais disponíveis, tamanho de *buffer* e número de modos de transmissão enquanto que se aumenta o número de dispositivos, este cenário pode ser tratado como um agrupamento de dispositivos no qual suas transmissões são coordenadas por uma parcela da estação base, ou seja, um cenário no qual a estação base poupe recursos para que seja possível conectar mais dispositivos numa mesma rede.

Na Figura 4.13, tem-se os valores médios de ocupação do *buffer* dos dispositivos, o algoritmo AR possui os menores valores em relação aos outros.

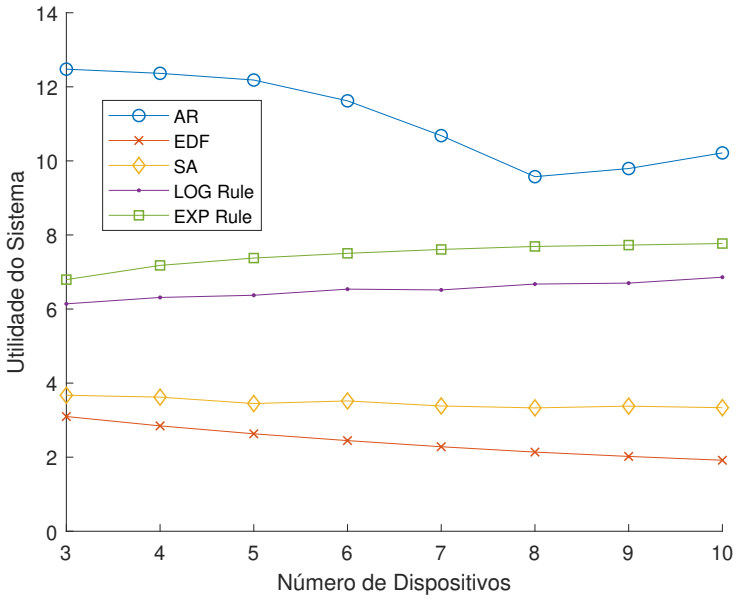
Figura 4.13 – Tamanho médio do *buffer* em relação ao número de dispositivos para o cenário 3




Fonte: Produzido pelo autor

Os valores de utilidade podem ser observados na Figura 4.14, nota-se que o algoritmo EDF possui os menores valores, inferiores até mesmo que o algoritmo aleatório, isso se dá pelo motivo de que a fila demora muitos intervalos de tempo para se atualizar pois a razão entre o número de dispositivos pelo número de canais disponíveis é bem maior que em relação ao cenário 2 isso faz com que a vazão seja inferior aos demais algoritmos. O algoritmo AR possui os maiores valores de utilidade.

Figura 4.14 – Utilidade do sistema em relação ao número de dispositivos para o cenário 3




Fonte: Produzido pelo autor

A potência consumida é mostrada na Figura 4.15, pode-se observar que o algoritmo EDF possui alto consumo de potência, pois ele prepara uma fila de prioridade e ao final dela, ao ser atualizada, pode ser que já exista pacotes a serem transmitidos e o algoritmo precisa esvaziar novamente os pacotes dos dispositivos, mantendo uma alta potência, mas com pouca eficiência, como se pode ver na Figura 4.17 no qual o EDF possui os maiores valores de pacotes perdidos sendo inferior somente ao algoritmo aleatório.

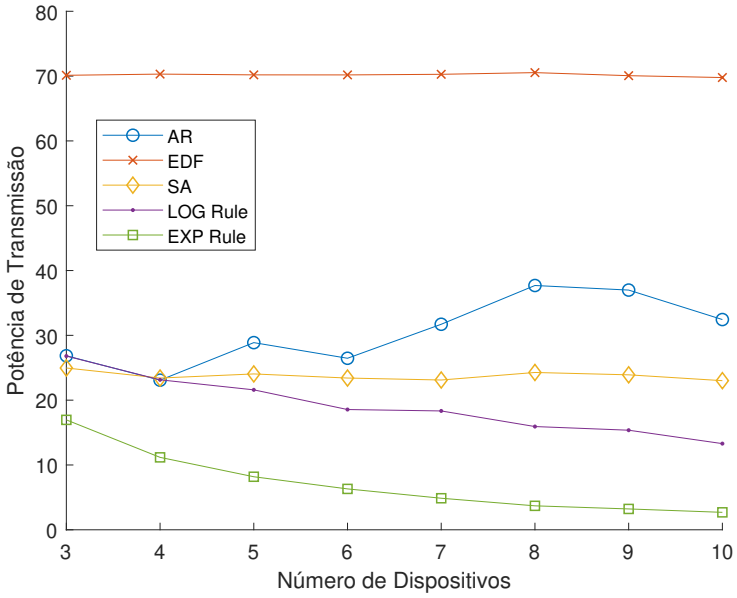
Na Figura 4.16 os valores de pressão dos *buffers* são mostrados, os valores crescem com o aumento do número de dispositivos e todos algoritmos apresentam comportamento muito parecido entre si, com o AR com os menores valores dentre eles.

A taxa de pacotes perdidos por pacotes transmitidos para os algoritmos utilizados na simulação são mostrados na Figura 4.17 onde se verifica que o algoritmo AR possui os menores valores de perda de pacotes enquanto que o EDF possui os maiores valores depois do algoritmo aleatório.

Na Figura 4.18 pode-se observar valores normalizados de vazão de pacotes, os baixos valores do algoritmo EDF, tão baixos quanto os da seleção aleatória, mostra que mesmo com o alto consumo de potência, não há vazão alta e ainda há muita perda de pacotes, evidenciando a falta de eficiência do algoritmo. O AR entretanto possui os maiores valores de vazão.

No cenário 3 observa-se uma mudança de comportamento principalmente do algoritmo EDF, com desempenho bem abaixo em relação ao algoritmo EDF no cenário 2,

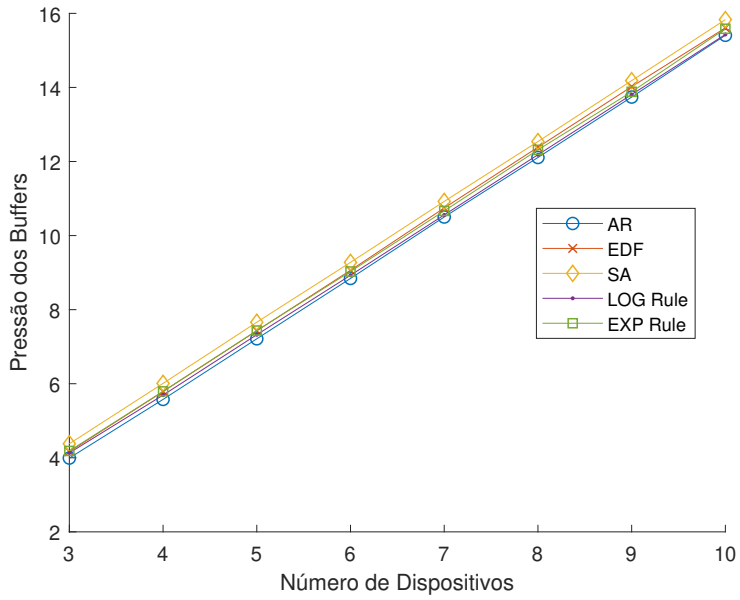
Figura 4.15 – Potência consumida em $mW$ pelo sistema em relação ao número de dispositivos para o cenário 3




Fonte: Produzido pelo autor

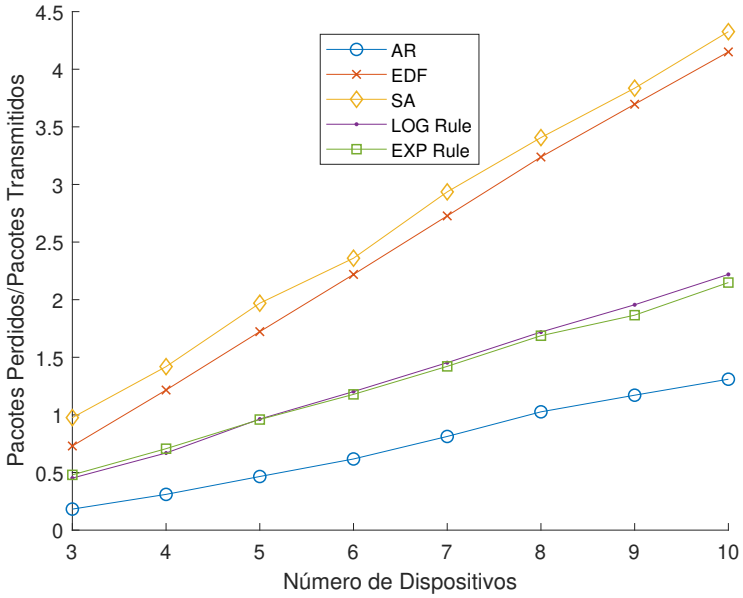
isso se dá por conta da limitação de número de canais que faz com que a fila criada seja maior em ralação ao cenário 2 ocasionando maiores quantidades de pacotes perdidos. Os algoritmos exponencial e logarítmico apresentam baixos valores de potência consumida, o que é uma grande qualidade no quesito de eficiência energética, entretanto os mesmos apresentam altos valores de pacotes perdidos também. O algoritmo AR possui os maiores valores de utilidade, mesmo não sendo tão econômico no consumo de potência, o algoritmo consegue compensar com um maiores valores de vazão de pacotes o que reflete na baixa taxa de pacotes perdidos.

Figura 4.16 – Pressão média dos *buffers* em relação ao número de dispositivos para o cenário 3



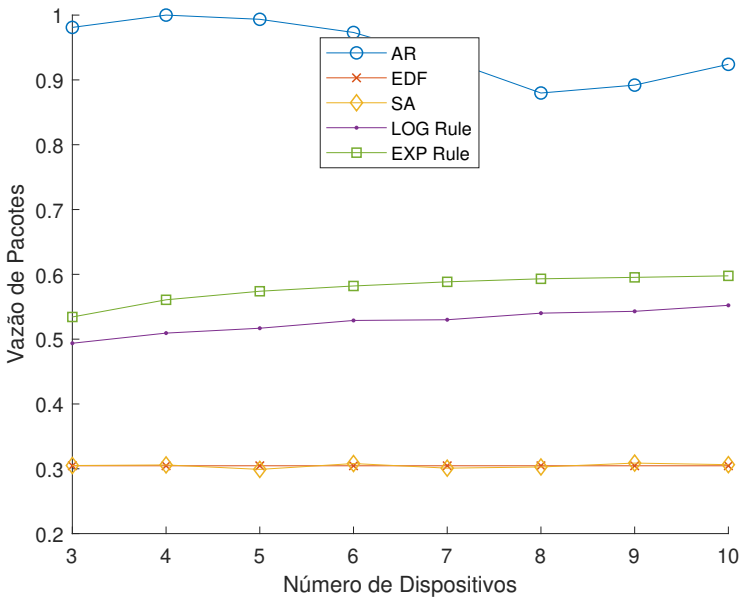

Fonte: Produzido pelo autor

Figura 4.17 – Taxa de pacotes perdidos por pacotes transmitidos em relação ao número de dispositivos para o cenário 3




Fonte: Produzido pelo autor

Figura 4.18 – Quantidade de pacotes transmitidos em relação ao número de dispositivos para o cenário 3



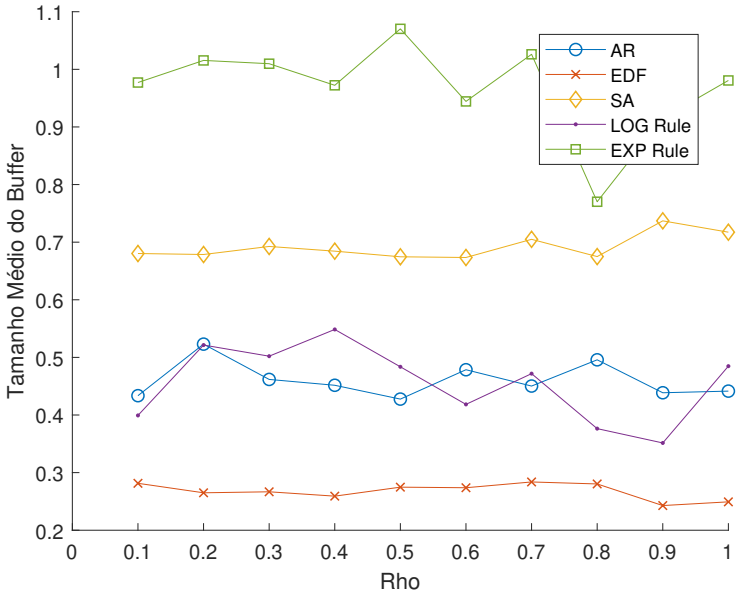

Fonte: Produzido pelo autor

### 4.1.4 Cenário 4 - Qualidade do Canal

O cenário 4 consiste no funcionamento dos algoritmos enquanto se aumenta a qualidade do canal de transmissão representado pelo parâmetro de Rayleigh $\rho$ que caracteriza a limiar da SNR. Neste cenário, utiliza-se a taxa de chegada com 0,5 pacotes/ms, $K = 4$ dispositivos, $L = 3$, 3 canais disponíveis com 2 estados de canal e $J = 2$.

Na Figura 4.19, pode-se visualizar os valores médios de ocupação do *buffer* em relação à limiar da SNR para os algoritmos considerados nas simulações. Nota-se que não há grandes alterações no tamanho médio do *buffer* ao variar o parâmetro $\rho$, indicando que a mudança do limiar da SNR não impacta diretamente nos valores de ocupação no *buffer*.

Figura 4.19 – Tamanho médio do *buffer* em relação ao parâmetro de Rayleigh para o cenário 4

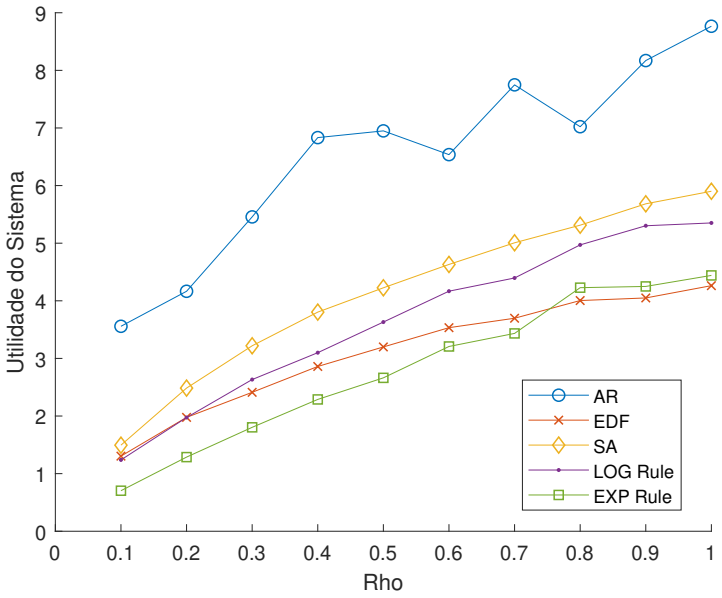



Fonte: Produzido pelo autor

Na Figura 4.20, visualizam-se os valores obtidos na simulação dos algoritmos para a utilidade do sistema em relação à limiar da SNR. Observa-se os valores crescem com a melhora na qualidade do canal o que é previsível pois há maior eficiência na potência consumida nos canais de transmissão que possuem maior qualidade. Os valores obtidos para a potência podem ser encontrados na Figura 4.21 comprovando a ocorrência de uma queda no consumo de potência de todos os algoritmos simulados como esperado.

Os valores de pressão dos *buffers* mostrados na Figura 4.22 e os valores da taxa de pacotes perdidos na Figura 4.23. Os resultados das Figuras das figuras 4.22 e 4.23 são condizentes, pois mostram que os algoritmos que exigem mais dos *buffers* tendem a perder mais pacotes, como o é o caso do algoritmo EXP. Ambas as Figuras mostram um

Figura 4.20 – Utilidade do sistema em relação ao parâmetro de Rayleigh para o cenário 4



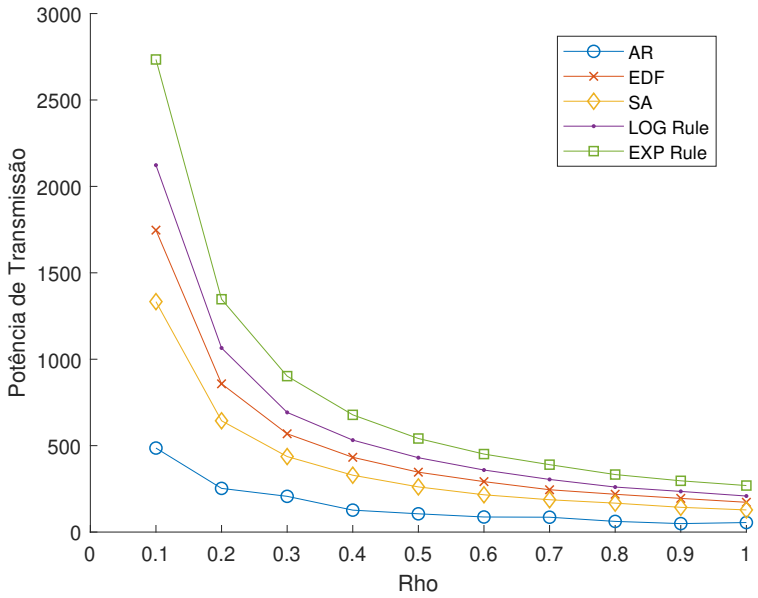

Fonte: Produzido pelo autor

comportamento aparentemente independente ao parâmetro de Rayleigh, indicando que não há uma relevância a ponto de relacionar os valores de pressão e de pacotes perdidos com o parâmetro $\rho$.

Os valores para a vazão de pacotes normalizada são visualizados na Figura 4.24. Observa-se uma maior quantidade de pacotes sendo transmitidos pelos algoritmos EXP e AR, onde destaca-se maiores valores de vazão obtidos pelo algoritmo EXP em relação ao algoritmo AR, denotando um comportamento enviesado em favor de um dispositivo, ao contrário do algoritmo AR que possui menores valores de taxa de perda de pacotes.

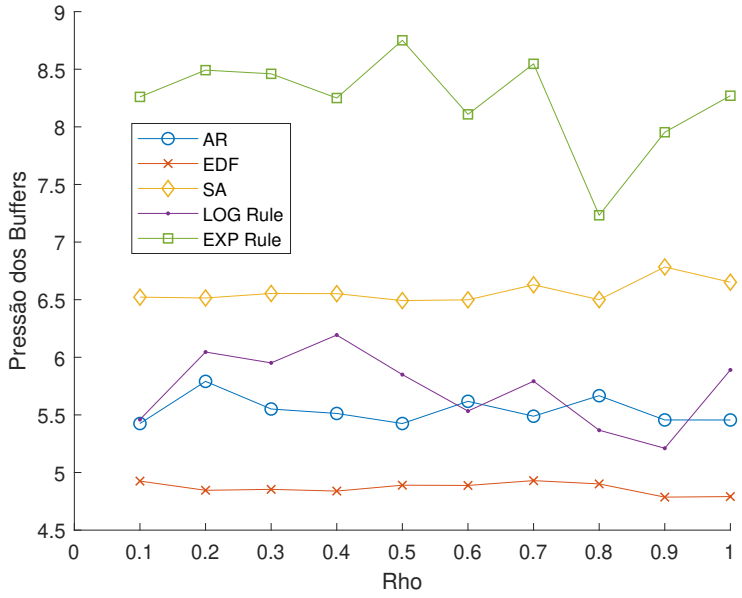
No cenário 4, o algoritmo proposto apresentou os maiores valores de utilidade, motivados pela redução do consumo de potência. O algoritmo EDF se comportou de maneira superior aos demais algoritmos em alguns aspectos, apresentando os melhores resultados em termos de pressão de *buffer* e perda de pacotes.

Figura 4.21 – Potência consumida em $mW$ pelo sistema em relação ao parâmetro de Rayleigh para o cenário 4



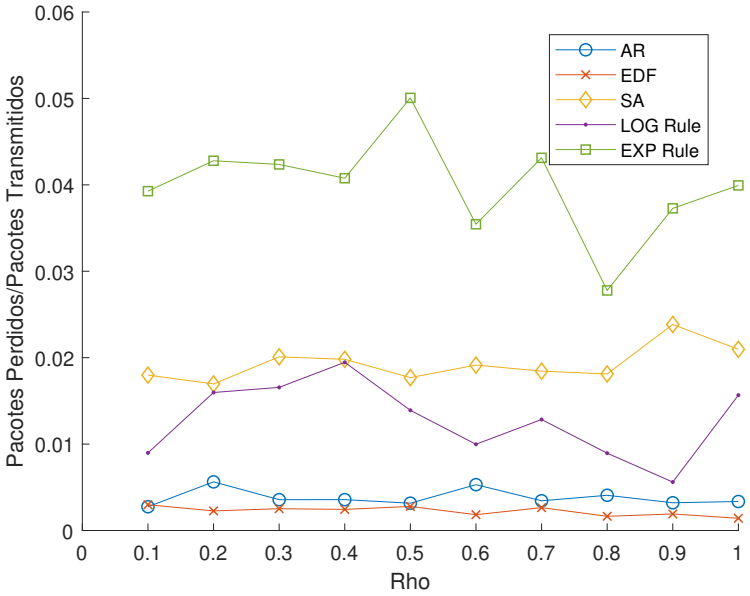

Fonte: Produzido pelo autor

Figura 4.22 – Pressão média dos *buffers* em relação ao parâmetro de Rayleigh para o cenário 4



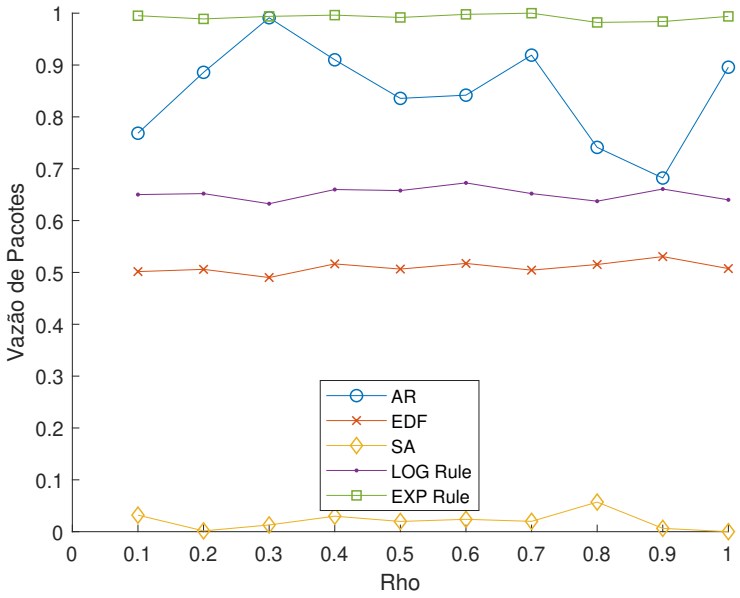

Fonte: Produzido pelo autor

Figura 4.23 – Taxa de pacotes perdidos por pacotes transmitidos em relação ao parâmetro de Rayleigh para o cenário 4




Fonte: Produzido pelo autor

Figura 4.24 – Quantidade de pacotes transmitidos em relação ao relação ao parâmetro de Rayleigh para o cenário 4




Fonte: Produzido pelo autor

## 4.2 Análise dos Resultados

De modo geral, o algoritmo AR possui os maiores valores de utilidade em todos os cenários simulados, mesmo sendo superado em alguns resultados pelo algoritmo EDF que se destaca pela baixa taxa de perda de pacotes, entretanto esse desempenho do algoritmo EDF não se mantém em todos os cenários simulados, no cenário 3, o algoritmo EDF possui os menores valores.

Observa-se que o baixo consumo energético do algoritmo AR é o que causa maior impacto nos altos valores de utilidade, entretanto, os baixos valores de potência consumida não comprometem o funcionamento do algoritmo, pois o algoritmo AR também apresenta os menores valores de perda de pacotes.

Em contrapartida, a maior desvantagem do algoritmo proposto é a própria limitação de atuação, pois devido a sua complexidade computacional há limite com relação ao número de dispositivos que podem ser considerados na simulação. Dessa forma, o algoritmo AR precisa ser executado em cenários com um limite no número de dispositivos. Já essa consideração não precisa ser feita para os outros algoritmos considerados neste trabalho.

O algoritmo AR se mostrou uma proposta satisfatória por possuir valores relativos altos de utilidade em todos os cenários simulados. Assim, o algoritmo AR provou ser uma opção viável e eficiente para alocação de recursos para sistemas IdC cognitivos.

# 5 Conclusão

Nesse trabalho, apresenta-se uma proposta de algoritmo utilizando aprendizagem por reforço baseada em Cadeia de Markov para realizar o escalonamento na transmissão de pacotes em um sistema de comunicação IdC cognitivo sem fio com múltiplos dispositivos. Para tal, adota-se uma cadeia de Markov para modelar os estados do sistema de comunicação e suas transições, fornecendo os parâmetros necessários para determinar ações de alocação de recursos de acordo com o Algoritmo Proposto 1.

O maior desafio da proposta está no treinamento do algoritmo AR pois sua complexidade computacional limita a sua aplicação a um número grande de dispositivos conectados a uma mesma estação base. Possíveis soluções para tentar amenizar esse problema de complexidade computacional seriam: utilizar técnicas de *clustering* para agregar dispositivos com características semelhantes, utilizar aproximações menos precisas para as estimativas da função $\mathbf{Q}$ e aumentar a capacidade de processamento e de armazenamento disponível.

Os resultados apresentados referentes às simulações computacionais de um sistema IdC cognitivo mostram que a aprendizagem por reforço tem um desempenho em geral superior em relação aos outros algoritmos de escalonamento considerados. O algoritmo AR proposto se destaca por apresentar valores de utilidade de sistema maiores nos três cenários simulados, onde o agente é treinado para priorizar a vazão de pacotes levando em consideração também o consumo de potência envolvido na transmissão e uma variável relacionada à pressão de *buffer*. Conclui-se portanto que o algoritmo AR proposto constitui uma opção viável e eficiente de alocação de recursos para o sistema modelado.

Uma sugestão de trabalho que poderia ser feito a partir do algoritmo proposto é a utilização de uma subestação base que se comportaria como uma estação base para um número limitado de dispositivos, o sistema completo seria formado desse conjunto de subestações gerenciado por um algoritmo de controle, assim, pode-se contornar a limitação do número de dispositivos IdC.

Uma outra sugestão de trabalho futuro seria o uso de técnicas para reduzir a complexidade computacional, como por exemplo o uso de técnicas de *clustering* para agrupar características semelhantes de dispositivos, ou agrupar dispositivos semelhantes, com o propósito de reduzir o processamento no treinamento do agente e permitir a simulação do algoritmo em cenários mais amplos, com mais dispositivos.

# Referências

ALAGOZ, O. et al. Markov decision processes: A tool for sequential decision making under uncertainty. *Medical Decision Making*, v. 30, n. 4, p. 474–483, 2010. PMID: 20044582. Disponível em: <https://doi.org/10.1177/0272989X09353194>. Citado na página 14.

CIRANI, S. et al. *Internet of things: architectures, protocols and standards*. [S.l.]: John Wiley & Sons, 2018. Citado na página 11.

GREWAL, J.; KRZYWINSKI, M.; ALTMAN, N. Markov models—markov chains. *Nature Methods*, Nature Publishing Group, v. 16, n. 8, p. 663–664, ago. 2019. ISSN 1548-7091. Citado na página 23.

HASEGAWA, S. et al. Performance evaluation of machine learning based channel selection algorithm implemented on iot sensor devices in coexisting iot networks. In: *2020 IEEE 17th Annual Consumer Communications Networking Conference (CCNC)*. [S.l.: s.n.], 2020. p. 1–5. Citado na página 20.

HAYKIN, S. *Digital Communication Systems*. Wiley, 2013. ISBN 9780471647355. Disponível em: <https://books.google.com.br/books?id=YGZXAAAACAAJ>. Citado na página 23.

JANG, S. et al. Research trends on deep reinforcement learning. *Electronics and Telecommunications Trends*, Electronics and Telecommunications Research Institute, v. 34, n. 4, p. 1–14, 2019. Citado na página 9.

KAELBLING, L. P.; LITTMAN, M. L.; CASSANDRA, A. R. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, Elsevier, v. 101, n. 1-2, p. 99–134, 1998. Citado na página 15.

KHURPADE, J. M.; RAO, D.; SANGHAVI, P. D. A survey on iot and 5g network. In: *2018 International Conference on Smart City and Emerging Technology (ICSCET)*. [S.l.: s.n.], 2018. p. 1–3. Citado 2 vezes nas páginas 11 e 19.

NAPARSTEK, O.; COHEN, K. Deep multi-user reinforcement learning for distributed dynamic spectrum access. *IEEE Transactions on Wireless Communications*, PP, p. 1–1, 11 2018. Citado na página 19.

PAPOULIS, A.; PILLAI, S. U. *Probability, Random Variables, and Stochastic Processes*. Fourth. Boston: McGraw Hill, 2002. ISBN 0071122567 9780071122566 0073660116 9780073660110 0071226613 9780071226615. Disponível em: <http://www.worldcat.org/search?qt=worldcat_org_all&q=0071226613>. Citado na página 12.

RUSSELL, S. J.; NORVIG, P. *Artificial Intelligence: a modern approach*. 3. ed. [S.l.]: Pearson, 2009. Citado 2 vezes nas páginas 16 e 17.

SADIQ, B.; MADAN, R.; SAMPATH, A. Downlink scheduling for multiclass traffic in lte. *EURASIP J. Wirel. Commun. Netw.*, Hindawi Limited, London, GBR, v. 2009, mar. 2009. ISSN 1687-1472. Disponível em: <https://doi.org/10.1155/2009/510617>. Citado na página 18.

SHAKKOTTAI, S.; STOLYAR, A. Scheduling algorithms for a mixture of real-time and non-real-time data in hdr. *Teletraffic Science and Engineering*, v. 4, 09 2001. Citado 2 vezes nas páginas 18 e 19.

SUTTON, R. S.; BARTO, A. G. *Reinforcement Learning: An Introduction*. Second. The MIT Press, 2018. Disponível em: <http://incompleteideas.net/book/the-book-2nd.html>. Citado na página 12.

TRIPATHY, B.; ANURADHA, J. *Internet of things (IoT): technologies, applications, challenges and solutions*. [S.l.]: CRC Press, 2017. Citado na página 11.

VERNON, D. *Artificial cognitive systems: A primer*. [S.l.]: MIT Press, 2014. Citado na página 21.

Wei, X. et al. Broad reinforcement learning for supporting fast autonomous iot. *IEEE Internet of Things Journal*, v. 7, n. 8, p. 7010–7020, 2020. Citado 2 vezes nas páginas 11 e 19.

XIONG, X. et al. Resource allocation based on deep reinforcement learning in iot edge computing. *IEEE Journal on Selected Areas in Communications*, PP, p. 1–1, 04 2020. Citado na página 20.

XIONG, Z. et al. Deep reinforcement learning for mobile 5g and beyond: Fundamentals, applications, and challenges. *IEEE Vehicular Technology Magazine*, v. 14, p. 44–52, 06 2019. Citado 2 vezes nas páginas 9 e 25.

ZAIDI, A. et al. *5G Physical Layer: principles, models and technology components*. [S.l.]: Academic Press, 2018. Citado 2 vezes nas páginas 22 e 28.

ZHANG, X. W. Y.; XU, Y. Energy-efficient resource allocation in uplink noma systems with deep reinforcement learning. In: *2019 11th International Conference on Wireless Communications and Signal Processing (WCSP)*. [S.l.: s.n.], 2019. p. 1–6. Citado na página 19.

ZHU, C. et al. 5g wireless networks meet big data challenges, trends, and applications. In: . [S.l.: s.n.], 2019. p. 1513–1516. Citado na página 9.