

**TROCA AUTOMÁTICA DE PROTOCOLOS MAC  
EMPREGANDO APRENDIZADO POR REFORÇO**



ANDRÉ VINÍCIUS GOMES SANTOS GONÇALVES

# TROCA AUTOMÁTICA DE PROTOCOLOS MAC EMPREGANDO APRENDIZADO POR REFORÇO

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais - Departamento de Ciência da Computação como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

ORIENTADOR: DANIEL FERNANDES MACEDO  
COORIENTADOR: LUIZ FILIPE MENEZES VIEIRA

Belo Horizonte  
Janeiro de 2019

© 2019, André Vinícius Gomes Santos Gonçalves.  
Todos os direitos reservados.

Gonçalves, André Vinícius Gomes Santos

G635t Troca automática de protocolos MAC empregando  
aprendizado por reforço / André Vinícius Gomes  
Santos Gonçalves. — Belo Horizonte, 2019  
xxii, 74 f. : il. ; 29cm

Dissertação (mestrado) — Universidade Federal de  
Minas Gerais - Departamento de Ciência da  
Computação

Orientador: Daniel Fernandes Macedo  
Coorientador: Luiz Filipe Menezes Vieira

1. Computação – Teses. 2. Redes de Computadores.  
3. Sistemas de comunicação sem fio. 4. Aprendizado do  
computador. I. Orientador. II. Coorientador.  
III. Título.

CDU 519.6\*22 (043)



UNIVERSIDADE FEDERAL DE MINAS GERAIS  
INSTITUTO DE CIÊNCIAS EXATAS  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

## FOLHA DE APROVAÇÃO

Troca automática de protocolos MAC empregando aprendizado por reforço

**ANDRÉ VINÍCIUS GOMES SANTOS  
GONÇALVES**

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:

  
PROF. DANIEL FERNANDES MACEDO - Orientador  
Departamento de Ciência da Computação - UFMG

  
PROF. LUIZ FÍLIPE MENEZES VIEIRA - Coorientador  
Departamento de Ciência da Computação - UFMG

  
PROFA. GISELE LOBO PAPP  
Departamento de Ciência da Computação - UFMG

  
PROF. LUIZ DA SILVA  
Engenharia Eletrônica e Elétrica - Trinity College Dublin

  
PROF. LUIZ HENRIQUE ANDRADE CORREIA  
Departamento de Ciência da Computação - UFLA

Belo Horizonte, 30 de janeiro de 2019.



*Dedico este trabalho à minha família e aos meus amigos, pelo apoio e incentivo.*



# Agradecimentos

Meus sinceros agradecimentos:

Aos meus pais e à minha família, pelo apoio e compreensão durante esse período;

Aos meus orientadores Daniel e Luiz Filipe, pelo suporte e orientação, guiando-me pelo mundo acadêmico-científico;

Aos meus companheiros de trabalho do laboratório Winet, em especial ao Henrique pelas experiências compartilhadas, ao Gilson, Lucas, Matheus C. e Racyus ("só vocês, galera!") pelos momentos de descontração, e ao Matheus N. e Marcos pelo suporte irrestrito durante o período de testes;

Aos meus amigos e namorada, por tornarem essa caminhada mais fácil por conta do suporte e incentivo;

À agência de fomento à pesquisa CAPES, por viabilizar financeiramente minha dedicação aos estudos;

Ao projeto FUTEBOL, por disponibilizar os equipamentos utilizados durante os experimentos;

A todos aqueles que contribuíram de certa forma para a concretização deste trabalho, direta ou indiretamente.



*“Keep it simple and stupid.”*  
(Marinha EUA, 1960)



# Resumo

Há uma crescente demanda por redes sem fio impulsionada pela popularização de dispositivos como laptops, smartphones e tablets, e por tecnologias emergentes como Internet das Coisas. Se por um lado a demanda é crescente, por outro, o uso eficiente de redes sem fio é desafiador. O ambiente sem fio é sensível, por exemplo, a variações de topologia da rede, obstáculos entre dispositivos, mobilidade, variações do meio de propagação como umidade do ar ou condições climáticas, etc. Não suficiente, o mesmo meio (o ar) é utilizado por todos os dispositivos, disputando entre si o uso do recurso.

Apesar da crescente demanda, dinamismo e complexidade de redes sem fio, alguns protocolos de comunicação são focados em situações específicas de utilização. Esse é o caso de protocolos MAC (do inglês *Media Access Control*). Por exemplo, classes de protocolos MAC baseadas em contenção tendem a ter melhor desempenho em redes sem fio com baixa disputa de acesso ao meio, tornando-se ineficientes a medida que a disputa aumenta. Por outro lado, protocolos MAC sem contenção possuem comportamento contrário, ineficientes em situações de baixa disputa. Devido ao dinamismo de redes sem fio, um único protocolo MAC é dificilmente o melhor durante todo o tempo. Uma alternativa é a troca do protocolo MAC de acordo com as condições da rede sem fio ao longo do tempo.

A proposta dessa dissertação ataca esse problema, consistindo em uma Subcamada auto-Organizável MAC (SOMAC) capaz de selecionar e trocar o protocolo MAC automaticamente ao longo do tempo. Há dois viabilizadores dessa solução. O primeiro consiste em rádios definidos por software, dispositivos que permitem a programação e a flexibilização da subcamada MAC. O segundo corresponde a técnicas de aprendizado de máquina, possibilitando a seleção autônoma de protocolos MAC. SOMAC faz uso dos dois, rádios definidos por software para implementação e validação, e aprendizado de máquina para o motor de seleção de protocolos.

SOMAC emprega o algoritmo de aprendizado por reforço Q-Learning e lida com o dinamismo de redes sem fio. O sistema periodicamente avalia métricas de rede e seleciona o protocolo MAC de melhor desempenho. SOMAC é implementado e validado

em ambientes de redes sem fio reais. Resultados experimentais sugerem que SOMAC é capaz de selecionar o melhor protocolo MAC em, pelo menos, 80% do tempo, chegando a atingir limiares de otimalidade de 90%. Além disso, resultados experimentais também retratam que SOMAC supera seus principais concorrentes em múltiplas situações, batendo o estado da arte.

**Palavras-chave:** Redes de Computadores, Subcamada MAC, Seleção de Protocolos MAC, Aprendizado de Máquina, Aprendizado por Reforço.

# Abstract

There is an increasing demand for wireless networks propelled by the popularization of mobile devices such as laptops, smartphones and tablets, and by emerging technologies such as the internet of things. If the demand is increasing on one hand, the efficient use of wireless networks is challenging on the other. For instance, wireless networks are at mercy of changes in the network topology, obstacles between devices, node's mobility, propagation medium fluctuations such as humidity or weather conditions etc. Further, the same medium is used by all devices, resulting in competition between nodes.

Despite the increasing demand, dynamism and complexity of wireless networks, some communication protocols focus on specific application scenarios, which is the case for MAC (Media Access Control) protocols. For example, contention-based MAC protocols tend to perform better in networks under low competition, becoming inefficient as competition rises. Conversely, contention-free MAC protocols tend to perform better when network competition is high. Since wireless networks are dynamic environments, a single MAC protocol is unlikely the best one all the time. A more efficient approach is to switch the MAC protocol according to the network conditions over time.

This dissertation tackles that problem, consisting of a Self-Organizing MAC sublayer (SOMAC) capable of automatically selecting and switching the MAC protocol over time. This solution is feasible for two reasons. The first is called software defined radios, consisting of programmable radios that allow the flexibilization of the MAC sublayer. The second corresponds to machine learning techniques, allowing an automatic MAC protocol selection. SOMAC deploys both: a) implementation and validation through software defined radios and b) a selection engine powered by machine learning.

SOMAC uses the reinforcement learning algorithm Q-Learning and copes with the dynamic nature of wireless networks. Our solution periodically assesses the wireless network performance through network metrics and selects the best MAC protocol. In addition, SOMAC is implemented and validated in real-world wireless networks. Our results indicate that SOMAC selects the best MAC protocol at a minimum rate of 80% of the time, reaching up to 90% of optimality. Furthermore, they also point that

SOMAC defeats its main competitors by a significant extent, outperforming the state of the art.

**Keywords:** Computer Networks, MAC Sublayer, MAC Protocol Selection, Machine Learning, Reinforcement Learning.

# Lista de Figuras

1.1	Camadas do modelo de referência OSI . . . . .	2
2.1	Resumo do protocolo de acesso ao meio usado em IEEE 802.11 [IEEE, 2016]	9
2.2	Funcionamento básico do TDMA . . . . .	10
2.3	Funcionamento do TDMA utilizado em SOMAC . . . . .	11
2.4	Charge sobre a criação de padrões . . . . .	12
2.5	Esquemático de um Rádio Definido por software ideal [Mitola, 1993] . . . .	13
2.6	GNU Radio Companion (GRC), [da Silva et al., 2015] . . . . .	14
2.7	Visão geral de um sistema de aprendizado por reforço, adaptado de [Sutton and Barto, 1998] . . . . .	18
4.1	Exemplo de topologia utilizada por SOMAC . . . . .	32
4.2	Arquitetura de software da subcamada MAC de SOMAC . . . . .	33
4.3	Diagrama de decisão . . . . .	34
4.4	Máquina de estados de SOMAC . . . . .	37
4.5	SOMAC: arquitetura de blocos na plataforma GNU Radio . . . . .	41
4.6	Interface de rede "TUNTAP" . . . . .	42
5.1	Fluxograma da análise numérica de SOMAC . . . . .	48
5.2	Disposição dos rádios USRP do <i>testbed</i> UFMG . . . . .	49
5.3	Configurações de rede . . . . .	50
5.4	$\epsilon$ -greedy, mapas de calor de $\eta$ em função de $\alpha$ , $\gamma$ e $\epsilon$ . . . . .	52
5.5	Softmax, mapas de calor de $\eta$ em função de $\alpha$ , $\gamma$ e $T$ . . . . .	53
5.6	UCB, mapas de calor de $\eta$ em função de $\alpha$ , $\gamma$ e $c$ . . . . .	53
5.7	Resultado geral de desempenho das estratégias de exploração com os hiperparâmetros ajustados . . . . .	54
5.8	Visão geral da avaliação experimental . . . . .	55
5.9	Topologias predefinidas . . . . .	58
5.10	Topologia predefinida com dois pontos de troca . . . . .	59

5.11	Desempenho geral para topologias predefinidas . . . . .	59
5.12	Topologia aleatória . . . . .	60
5.13	Recompensa baseada na latência média . . . . .	61
5.14	Comparação com o estado da arte em topologias predefinidas . . . . .	62
5.15	Comparação com o estado da arte em topologias com 2 PTs e aleatória . .	62
5.16	Comparação com o estado da arte em configurações com 2 PTs e aleatória	63
5.17	Comparação com o estado da arte em configurações aleatórias . . . . .	64

# Lista de Tabelas

3.1	Comparação entre trabalhos relacionados (Li - limiar; MM - modelagem matemática; Fu - lógica fuzzy; AM - aprendizado de máquina; AR - aprendizado por reforço) . . . . .	28
4.1	Métricas de desempenho disponíveis para seleção do protocolo MAC . . . .	33
4.2	Hiperparâmetros por estratégia de exploração . . . . .	39



# Sumário

<b>Agradecimentos</b>	<b>ix</b>
<b>Resumo</b>	<b>xiii</b>
<b>Abstract</b>	<b>xv</b>
<b>Lista de Figuras</b>	<b>xvii</b>
<b>Lista de Tabelas</b>	<b>xix</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Contextualização . . . . .	1
1.1.1 Redes sem fio . . . . .	2
1.1.2 Aprendizado de máquina . . . . .	3
1.2 Motivação e Objetivo . . . . .	4
1.3 Contribuições . . . . .	4
1.4 Organização . . . . .	5
<b>2 Referencial teórico</b>	<b>7</b>
2.1 Subcamada MAC . . . . .	7
2.1.1 CSMA/CA . . . . .	8
2.1.2 TDMA . . . . .	10
2.1.3 Escolha do protocolo MAC . . . . .	11
2.2 Rádios Definidos por software . . . . .	13
2.2.1 GNU Radio . . . . .	14
2.2.2 Dispositivos SDR . . . . .	15
2.3 Aprendizado por Reforço . . . . .	17
2.3.1 Q-Learning . . . . .	18
2.3.2 Q-Learning em ambientes não estacionários . . . . .	21

2.4	Resumo . . . . .	22
<b>3</b>	<b>Trabalhos relacionados</b>	<b>23</b>
3.1	Protocolos MAC híbridos . . . . .	23
3.2	Protocolos MAC adaptáveis . . . . .	25
3.3	Troca de protocolos MAC . . . . .	26
3.4	Resumo . . . . .	29
<b>4</b>	<b>SOMAC: Subcamada auto-Organizável MAC</b>	<b>31</b>
4.1	SOMAC . . . . .	31
4.1.1	Arquitetura de SOMAC . . . . .	31
4.1.2	Mecanismo de seleção . . . . .	35
4.2	Implementação . . . . .	40
4.2.1	GNU Radio . . . . .	40
4.2.2	Conformidade com a norma IEEE 802.11 . . . . .	43
4.3	Resumo . . . . .	45
<b>5</b>	<b>Avaliação</b>	<b>47</b>
5.1	Análise e ajuste de hiperparâmetros . . . . .	47
5.1.1	Metodologia . . . . .	47
5.1.2	Resultados . . . . .	51
5.1.3	Discussão . . . . .	52
5.2	Avaliação experimental . . . . .	54
5.2.1	Metodologia . . . . .	55
5.2.2	Resultados . . . . .	56
5.3	Resumo . . . . .	64
<b>6</b>	<b>Conclusões</b>	<b>65</b>
	<b>Referências Bibliográficas</b>	<b>69</b>

# Capítulo 1

## Introdução

Este capítulo apresenta as motivações desta dissertação. A primeira seção contextualiza o escopo do problema. A segunda apresenta a motivação e o objetivo dessa dissertação. Por fim, as seções remanescentes descrevem as principais contribuições desse trabalho assim como a organização dos demais capítulos.

### 1.1 Contextualização

O número de dispositivos sem fio tem crescido. A popularização de dispositivos como *laptops*, *smartphones* e *tablets* é um dos principais propulsores desse crescimento. Somente no Brasil, o número de dispositivos móveis era de 280 milhões em 2017 segundo estimativas de [Lobo, 2017], sendo o número de *smartphones* próximo de 1 por habitante. Além disso, conceitos emergentes como o de Internet das Coisas (ou *Internet of Things*) tornam esse crescimento ainda mais expressivo. Um relatório divulgado por [Evans, 2011] especula que o número desses dispositivos chegue a 50 bilhões até 2020 em escala global. Desses, a grande maioria será de dispositivos móveis conectados às redes sem fio.

Não obstante à diversidade e à quantidade de dispositivos móveis, o ambiente sem fio é passível de uma série de problemas não encontrados em redes cabeadas. Por exemplo, características como umidade do ar, topologia local, obstáculos entre dispositivos, mobilidade etc. interferem no desempenho de redes sem fio. Além disso, o mesmo meio (o ar) é utilizado por todos os dispositivos da rede, disputando entre si o uso do recurso. Esses fatores dificultam o funcionamento eficiente de redes sem fio.

Em contrapartida à alta demanda, diversidade e complexidade de redes sem fio, algumas iniciativas têm sido propostas nos últimos anos. Por exemplo, [Moura et al., 2015] propõe uma plataforma de flexibilização para administração e configuração de

redes sem fio. Já SON (*Self-Organizing Networks*) é uma iniciativa de automação dos processos de configuração, otimização e manutenção de redes móveis de quarta e quinta geração [Nohrborg, 2018]. Similarmente, esta dissertação apresenta uma solução na direção de auto-organização de redes sem fio.

### 1.1.1 Redes sem fio

Redes de comunicação são usualmente baseadas no modelo de referência OSI (do inglês *Open System Interconnection*). Esse modelo é estruturado em camadas como ilustrado pela figura 1.1. Redes sem fio se diferenciam das demais basicamente pelas duas camadas inferiores. A camada 1 (camada física ou simplesmente PHY) está relacionada diretamente ao meio no qual a informação é transmitida (i.e. ar em redes sem fio). O esquema de modulação é um exemplo de função atribuída à camada PHY. Já o acesso ao meio é uma das funções atribuídas à camada 2 (camada de enlace). A orquestração do acesso é feita via protocolos de acesso ao meio, usualmente referidos como protocolos MAC (do inglês *Medium Access Control*). Esses coordenam quando o nó sem fio deve transmitir e quando deve permanecer em estado de espera ou escuta, aguardando o momento apropriado para a transmissão.



Figura 1.1: Camadas do modelo de referência OSI

A escolha de um protocolo MAC apropriado é importante para o funcionamento eficiente da rede sem fio. Por exemplo, protocolos baseados em reserva de recursos (e.g. tempo ou frequência) podem resultar em uma rede sem fio ociosa. Nesse caso, nós sem fio deixam de transmitir por não usarem recursos reservados a outros nós, mesmo quando esses últimos não possuem dados para transmissão. Assim, recursos reservados e não utilizados são desperdiçados, podendo resultar em ociosidade da rede sem fio. Por outro lado, protocolos baseados em disputa (i.e. sem reserva de recursos)

podem resultar em múltiplas colisões<sup>1</sup> em situações de alta demanda, degradando o desempenho da rede.

A abordagem apropriada de acesso ao meio em uma certa rede pode sofrer variações ao longo do tempo. Por exemplo, uma rede com um ou poucos nós sem fio pode ter melhor desempenho com protocolos MAC baseados em disputa. Contudo, isso tende a mudar à medida que novos nós são conectados à rede. Como redes sem fio permitem mobilidade, a inserção ou saída de nós sem fio é algo comum ao longo do tempo.

Adicionalmente, a quantidade de nós sem fio não é o único fator que impacta no desempenho de protocolos MAC. Maiores informações sobre o funcionamento de protocolos MAC e as respectivas limitações são fornecidas na seção 2.1 do próximo capítulo. Por ora, é importante mencionar que técnicas como aprendizado de máquina podem ser empregadas para criação de mecanismos automáticos de controle e organização da camada. Assim, as políticas de acesso ao meio da camada de enlace são alteradas sempre que necessário, melhorando o desempenho da rede sem fio.

### 1.1.2 Aprendizado de máquina

Aprendizado de máquina é um campo associado com Inteligência Artificial. Assim como definições de inteligência e aprendizado são generalizadas, definições de aprendizado de máquina também apresentam variações. De acordo com Arthur Samuel (1959), aprendizado de máquina é o “campo de estudo que permite que computadores tenham a habilidade de aprender sem que sejam explicitamente programados” (em tradução livre), enquanto [Murphy, 2012] define como um conjunto de métodos capazes de detectar padrões automaticamente a partir de dados, usando-os para predição futura de informação ou outros tipos de decisão sob incerteza. Já Tom Mitchel define como:

"Diz-se que um programa de computador aprende, a partir da experiência  $E$  em relação a uma classe de tarefas  $T$  e uma medida de desempenho  $P$ , se seu desempenho nas tarefas de  $T$ , medido por  $P$ , melhora com a experiência  $E$ ."

(Tom Michel, 1997 - em tradução livre)

Tipicamente, técnicas de aprendizado de máquina consistem na tomada de decisões feitas por computadores sem a intervenção ou assistência humana. Várias aplicações são baseadas nesse conceito atualmente. Por exemplo, uma aplicação popular é o

---

<sup>1</sup>Quando dois ou mais nós sem fio transmitem ao mesmo tempo, interferindo na transmissão um do outro.

filtro de correio eletrônico, onde aprendizado de máquina é utilizado para classificar e filtrar automaticamente mensagens de *email* indesejadas [Murphy, 2012].

De forma similar, aprendizado de máquina pode ser utilizado em redes sem fio no âmbito de automatizar decisões a fim de melhorar o desempenho da rede. Como mencionado por [Boutaba et al., 2018], alguns exemplos são: (1) seleção de potência e frequência adequadas para uma determinada rede sem fio; (2) roteamento de tráfego via a predição de demanda ao longo do tempo; ou (3) detecção de problemas devido à condições anormais da rede.

Outra possível aplicação é a definição das políticas de acesso ao meio. A seção 2.1 do próximo capítulo descreve especificidades de diferentes protocolos MAC, cada qual com sua política de acesso ao meio. Nesse escopo, técnicas de aprendizado de máquina podem selecionar protocolos MAC que correspondem às políticas de acesso ao meio adequadas à situação da rede sem fio em um dado momento, alterando o protocolo quando necessário.

## 1.2 Motivação e Objetivo

Esta dissertação é motivada pelo problema de seleção do protocolo MAC, descrito em maiores detalhes na seção 2.1.3 do próximo capítulo. Resumidamente, protocolos MAC são usualmente específicos a condições da rede sem fio. Por outro lado, redes sem fio são dinâmicas, cujas características são variáveis ao longo do tempo. Assim, um só protocolo MAC é dificilmente o mais adequado todo o tempo. Portanto, o objetivo deste trabalho é resumido na seguinte questão:

Dado o atual protocolo MAC e uma métrica de desempenho  $m$ , onde  $m$  pode ser uma única métrica ou um conjunto de métricas de rede agregadas em um valor escalar, deve-se manter ou alterar o protocolo MAC para otimizar a métrica  $m$ ?

## 1.3 Contribuições

Tendo em vista o objetivo definido para a dissertação, propomos um sistema que emprega aprendizado de máquina, mais especificamente aprendizado por reforço, para modificar dinamicamente o protocolo MAC sendo empregado.

Apesar de existirem outros mecanismos de decisão, a opção por aprendizado de máquina é baseada, principalmente, em dois fatores. Primeiro, soluções determinísticas são inviáveis devido ao grande número de fatores que podem influenciar no

funcionamento de redes sem fio [Ferrand et al., 2016]. Segundo, redes sem fio são não estacionárias [Ferrand et al., 2016]. Isso exige que o mecanismo de seleção seja capaz de atualizar-se ao longo do tempo, impossibilitando soluções dependentes de intervenção humana.

Assim, as principais contribuições do trabalho são:

- A proposta de um sistema auto-organizável de acesso ao meio, capaz de selecionar o melhor protocolo MAC de acordo com o estado da rede sem fio;
- A implementação desse sistema em rádios definidos por software (capítulo 2, seção 2.2) e a disponibilização, de forma aberta, desse sistema auto-organizável<sup>2</sup>;
- A avaliação do sistema proposto em redes sem fio reais.

## 1.4 Organização

O restante desta dissertação é organizado como segue. O segundo capítulo apresenta conceitos base, cobrindo protocolos MAC, rádios definidos por software e aprendizado por reforço. O capítulo 3 lista os principais trabalhos relacionados encontrados na literatura. Em seguida, o capítulo 4 discute a arquitetura da solução proposta e o modelo de aprendizado por reforço nela utilizado. Já o quinto capítulo discorre sobre a metodologia de avaliação e os resultados da execução da arquitetura proposta em redes sem fio reais. Por fim, o capítulo 6 apresenta as considerações finais e os trabalhos futuros.

---

<sup>2</sup>Código fonte disponível em <https://github.com/avgsg/gr-somac.git>



# Capítulo 2

## Referencial teórico

Este capítulo discute alguns conceitos básicos relacionados à dissertação. A primeira seção apresenta as atribuições da subcamada MAC e os principais protocolos MAC em uso em redes sem fio. A segunda seção apresenta conceitos básicos de rádios definidos por software. Por fim, a terceira seção descreve aprendizado por reforço, com foco no algoritmo Q-Learning.

### 2.1 Subcamada MAC

Em redes sem fio, o meio de transmissão é compartilhado pelos dispositivos conectados à rede. Portanto, é necessário que haja mecanismos de coordenação para possibilitar que todos os dispositivos possam transmitir de forma eficiente, evitando colisões (dois ou mais dispositivos transmitindo ao mesmo tempo) ou ociosidade (intervalos de inatividade mesmo quando dispositivos têm dados para transmissão). A coordenação do acesso ao meio é feita pelos protocolos MAC.

Há duas grandes classes de protocolos MAC [Busch et al., 2004]: com contenção e sem contenção. Protocolos MAC com contenção não possuem coordenação estrita de acesso ao meio nem reserva de recursos [Busch et al., 2004]. Os nós de rede disputam o uso do meio direta e localmente. Exemplos de protocolos MAC com contenção são: ALOHA (*pure* e *slotted*), CSMA (*Carrier-Sense Multiple Access*) e variações.

Protocolos MAC sem contenção possuem forte coordenação de acesso ao meio e/ou reserva de recursos [Busch et al., 2004]. Um exemplo é o protocolo TDMA (*Time Division Multiple Access*), cuja coordenação de acesso ao meio é feita pela reserva de janelas de transmissão para cada nó da rede. Outros recursos também podem ser reservados. Por exemplo, o protocolo FDMA (*Frequency Division Multiple Access*)

reserva uma frequência específica para cada nó enquanto o protocolo CDMA (*Code Division Multiple Access*) reserva códigos para codificação de um canal de transmissão.

Os protocolos MAC CSMA/CA e TDMA são de grande importância para essa dissertação. Consequentemente, ambos são discutidos nas próximas seções.

### 2.1.1 CSMA/CA

O protocolo CSMA/CA é uma variação da família de protocolos CSMA (*Carrier-Sense Multiple Access*). Em CSMA, cada nó detecta se o meio já está sendo utilizado por outros nós antes de cada transmissão [Kurose and Ross, 2012]. Caso o meio esteja ocioso, o nó transmite; caso contrário, não transmite e espera por uma nova oportunidade. A variação CSMA/CA (*Carrier-Sense Multiple Access with Collision Avoidance*) diz respeito ao fato do protocolo apenas evitar colisões mas não detectá-las diretamente caso ocorram, diferentemente da variação CSMA/CD (*Carrier-Sense Multiple Access with Collision Detection*), capaz de detectar colisões durante a transmissão [Kurose and Ross, 2012].

CSMA/CA é a variação de protocolos CSMA mais popular em redes sem fio. A detecção de colisões é uma tarefa difícil em redes sem fio [Kurose and Ross, 2012]. Primeiro, a energia do sinal recebido é significativamente pequena quando comparada ao sinal transmitido. Isso dificulta a detecção de sinais alheios durante a transmissão, já que o próprio sinal transmitido irá mascarar quaisquer outros sinais. Segundo, nem todos os nós são visíveis entre si em redes sem fio. Como consequência, problemas como o nó escondido<sup>1</sup> impossibilitam a detecção de colisões simultaneamente à transmissão [Kurose and Ross, 2012]. Assim, mecanismos para evitar colisões sem detectá-las são mais viáveis em redes sem fio e são empregados em padrões populares de comunicação.

O protocolo MAC CSMA/CA é parte fundamental do conjunto de especificações de redes sem fio IEEE 802.11 [IEEE, 2016]. Adicionalmente ao processo de detecção do uso do meio, IEEE 802.11 também utiliza a transmissão de quadros de confirmação para saber se a transmissão foi bem sucedida ou não. A figura 2.1 resume o processo. Ao receber um novo quadro para transmissão, o nó detecta se o meio já está sendo utilizado (ou não) através da escuta do meio por um tempo DIFS (do inglês *DCF Interframe Space*). Caso o meio esteja ocupado, o nó aguarda um intervalo de tempo aleatório (*Backoff* aleatório [IEEE, 2016]) e repete a verificação. O *backoff* aleatório é delimitado por uma janela de contenção (CW) que cresce exponencialmente (potência de 2) à medida que o meio é dito ocupado, tendendo a aumentar o tempo de espera em casos de tentativas sucessivas e mal sucedidas de transmissão. Caso o meio esteja

---

<sup>1</sup>Do inglês *hidden node problem*: [https://en.wikipedia.org/wiki/Hidden\\_node\\_problem](https://en.wikipedia.org/wiki/Hidden_node_problem).

livre, o quadro é enviado à camada física (PHY) para transmissão. Nesse caso, o nó aguarda uma mensagem de confirmação de recebimento por parte do destinatário. Uma retransmissão (ReTx) é engatilhada caso a mensagem de confirmação não seja recebida após um intervalo de tempo pré-determinado (SIFS – do inglês *Short Interframe Space*), repetindo-se o mesmo processo.

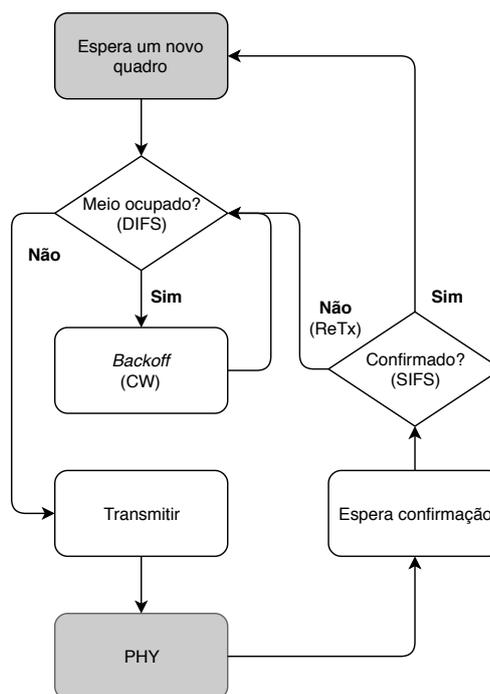


Figura 2.1: Resumo do protocolo de acesso ao meio usado em IEEE 802.11 [IEEE, 2016]

Essa abordagem tende a funcionar bem em cenários de baixa contenção (ou baixa disputa). Nesse caso, o nó tende a não cair em períodos de espera aleatória (*Backoff*) e realiza a transmissão tão pronto quanto a chegada de um novo quadro. Por outro lado, a transmissão pode ser retardada significativamente em ocasiões de alta disputa, fazendo com que o nó caia no estado de *backoff* mais frequentemente [Ziouva and Antonakopoulos, 2002]. Também é notório lembrar que CSMA/CA não garante a ausência de colisões, apenas as reduz. Portanto, a alta disputa do meio também pode resultar em maior incidência de colisões, exigindo retransmissões e degradando o desempenho da rede. Nesse caso, protocolos MAC sem contenção podem ser uma melhor alternativa.

### 2.1.2 TDMA

O protocolo de acesso ao meio TDMA pertence à classe de protocolos sem contenção. Nele, cada nó possui uma janela de tempo (ou simplesmente *slot*) exclusiva para transmissão. Assim, não há disputa direta entre os nós da rede para utilização do meio. Por outro lado, tem-se a necessidade de coordenação entre toda a rede, de modo que cada nó saiba exatamente quando pode transmitir.

Um exemplo de utilização do TDMA são redes móveis 2G, operando de acordo com o esquema GSM FDMA/TDMA [Kurose and Ross, 2012]. Nesse esquema, dispositivos são divididos em grupos, e cada grupo opera em uma frequência específica (componente FDMA). Dentro de um mesmo grupo, cada dispositivo recebe um *slot* de tempo diferente para transmissão de dados (componente TDMA).

A figura 2.2 ilustra o princípio de funcionamento do protocolo TDMA. Ela ilustra uma rede com 4 dispositivos, identificados com números de 1 a 4. Observe que cada nó possui um *slot* específico que pode ser usado para transmissão caso necessário. O ciclo de transmissão se repete ao longo do tempo.

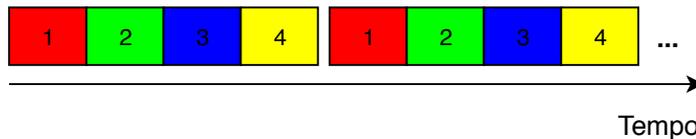


Figura 2.2: Funcionamento básico do TDMA

A principal vantagem do protocolo TDMA é a baixa incidência de colisões, uma vez que cada nó possui um *slot* de tempo específico para transmissão [Demirkol et al., 2006]. Isso favorece cenários com maior número de nós e alta demanda uma vez que evita retransmissões causadas por colisões. Por outro lado, colisões ainda podem ocorrer por dessincronizações entre os nós, acarretando em sobreposição de *slots*. A dificuldade de manter-se os nós sincronizados é, inclusive, uma das principais desvantagens do protocolo TDMA [Demirkol et al., 2006]. Outra desvantagem é a presença de *slots* ociosos. Um *slot* ocioso é um *slot* reservado para um determinado nó mas não utilizado. Suponha que os nós 1 e 2 possuem dados enfileirados para a transmissão enquanto os nós 3 e 4, não (em referência à figura 2.2). Nesse caso, os recursos reservados aos nós 3 e 4 são desperdiçados, já que não podem ser utilizados pelos demais, degradando o desempenho da rede.

Algumas variações do protocolo TDMA lidam com *slots* ociosos. Ainda assim, desvantagens podem ser apontadas. Por exemplo, o trabalho de [ITU-R, 1997] dinamicamente aloca *slots* de acordo com a demanda de tráfego. Nele, há um canal de controle exclusivo, permitindo a alocação de *slots* ociosos a nós com maior demanda

de tráfego. Por outro lado, a sobrecarga necessária para o controle consome recursos que poderiam ser alocados para comunicação.

Outro exemplo é descrito no trabalho de [Cordeiro et al., 2018]. Há dois períodos que dividem o ciclo do protocolo TDMA: 1) alocação e 2) comunicação. Por sua vez, o período de alocação consiste em duas etapas. Na primeira, os nós da rede informam sobre a utilização dos respectivos *slots* de tempo. Na segunda, os nós que requisitaram *slots* são informados sobre a ordem de transmissão. Similarmente ao trabalho de [ITU-R, 1997], a sobrecarga de mensagens controle consome recursos que poderiam ser alocados para comunicação, especialmente a primeira etapa, que deve durar tempo suficiente para todos os nós da rede informarem sobre a utilização ou não dos respectivos *slots*.

Neste trabalho, adotamos uma variação do protocolo TDMA de [Cordeiro et al., 2018]. A figura 2.3 ilustra esse processo. O período de alocação é utilizado apenas para informar os nós da rede sobre a ordem de transmissão (*slot* ORD). A primeira etapa de [Cordeiro et al., 2018] é substituída por um contador temporal, onde nós sem fio são excluídos do ciclo de transmissão por inatividade depois de 30 segundos. O último *slot* de cada ciclo é destinado à novos nós, sendo esses adicionados ao próximo ciclo. Apesar dessa abordagem permitir a presença de *slots* ociosos durante um intervalo de tempo maior que [Cordeiro et al., 2018], ela possui menor sobrecarga de mensagens de controle quando a rede possui alta disputa (i.e. todos os nós com dados para transmissão), beneficiando o desempenho em situações que o algoritmo TDMA tende a ter melhor desempenho.



Figura 2.3: Funcionamento do TDMA utilizado em SOMAC

### 2.1.3 Escolha do protocolo MAC

A escolha do protocolo MAC deve levar em consideração vantagens e desvantagens de cada protocolo. Como visto acima, CSMA/CA e TDMA possuem pontos positivos e negativos de acordo com o estado da rede sem fio.

CSMA/CA é uma boa alternativa em redes sem fio com baixa disputa como apontado na seção 2.1.1. Particularmente, o protocolo tende a ter um melhor desempenho quando há poucos nós presentes na rede sem fio, evitando colisões e longos períodos de

espera (*backoff*). Por outro lado, TDMA é uma boa escolha em cenários de alta disputa mas apresenta degradação de desempenho na presença de *slots* ociosos ou devido à sobrecarga de mecanismos de controle (seção 2.1.2).

Peculiaridades não são exclusividade dos protocolos CSMA/CA e TDMA. Outras propostas de protocolos MAC podem ser encontradas na literatura, cada qual com seus respectivos cenários de aplicação. [Choudhury et al., 2006] apresenta DMAC, um protocolo MAC que explora a diretividade de antenas direcionais, porém é mais sensível ao problema do nó escondido; [Guo et al., 2012] propõe um protocolo baseado no TDMA focado em ambientes com alta mobilidade, contudo o *overhead* necessário compromete o desempenho em situações de média e baixa mobilidade; [Al-meffleh, 2009] aponta as vantagens do protocolo DCF em termos de equidade de acesso ao meio, todavia podendo sofrer com longos atrasos dependendo do nível de contenção da rede.

A figura 2.4 ilustra de forma cômica a busca por padrões globais. De forma análoga, proposições de novos protocolos MAC parecem resultar em novos protocolos com características específicas de cenários de aplicação, trazendo ambos vantagens e desvantagens como exemplificado acima.



Figura 2.4: Charge sobre a criação de padrões

Fonte: <https://xkcd.com/927/>

Se por um lado protocolos MAC tendem a ser específicos de acordo com o estado da rede, por outro, redes sem fio são dinâmicas. Usualmente, número de dispositivos conectados, topologia, demanda, mobilidade etc. são fatores variantes ao longo do tempo em redes sem fio. Nesse contexto, uma melhor abordagem seria a utilização de diferentes protocolos MAC em uma rede sem fio, alterando-os de acordo com o estado da rede.

## 2.2 Rádios Definidos por software

Rádio Definido por software (do inglês *Software Defined Radio*, ou simplesmente SDR) é um rádio capaz de acomodar uma grande variedade de funções de rádio frequência (RF) em software [Mitola, 2000], funções essas usualmente implementadas em hardware em rádios tradicionais. O arcabouço de funções implementadas em software pode incluir: geração de sinais, modulação e demodulação de sinais, controle de erros etc.

A figura 2.5 ilustra a proposta de um rádio definido por software ideal proposto por [Mitola, 1993]<sup>2</sup>. É importante ressaltar que componentes de hardware ainda estão presentes como, por exemplo, conversores analógicos/digitais e vice-versa (A/D e D/A). Rádios definidos por software representam a expansão de funcionalidades ora implementadas em hardware para implementação em software.

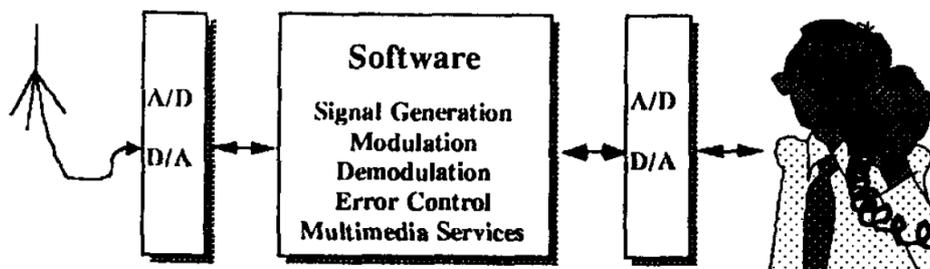


Figura 2.5: Esquemático de um Rádio Definido por software ideal [Mitola, 1993]

Implementações em software geram maior flexibilidade em relação à rádios tradicionais. Um mesmo rádio pode operar em diferentes tipos de redes sem fio, bastando alterar o software a ser executado. Além disso, o rádio pode operar de acordo com diferentes formatos de sinais e explorar protocolos de comunicação, adaptando-os sempre que conveniente [Mitola, 1993]. Por outro lado, rádios definidos por software possuem desempenho inferior se comparados à rádios tradicionais [da Silva et al., 2015]. Apesar de pouco flexíveis, implementações em hardware são otimizadas para um conjunto específico de tarefas, resultando em melhor desempenho.

Atualmente, soluções comerciais de rádios definidos por software estão disponíveis no mercado. Um exemplo é a empresa irlandesa *Software Radio Systems Limited* (SRS), que desenvolve ferramentas voltadas para as camadas L1, L2 e L3 de redes móveis baseadas em rádios definidos por software [Limited, 2016]. Outro exemplo é a empresa britânica *Clutch Space Systems* (CSS). A CSS utiliza rádios definidos por

<sup>2</sup>Dr. Joseph Mitola III é conhecido popularmente como "pai do rádio definido por software" e tem produzido uma série de trabalhos relacionados ao tema ao longo dos últimos anos, [https://dblp.uni-trier.de/pers/hd/m/Mitola\\_III:Joseph](https://dblp.uni-trier.de/pers/hd/m/Mitola_III:Joseph).

software para criação de soluções de comunicação via satélite [Systems, 2018]. Há também iniciativas de operadoras de telefonia como a Vodafone para a utilização de SDR [Bicheno, 2017], [Vodafone, 2018]. Adicionalmente, o *Wireless Innovation Forum* divulgou em 2011 uma nota constatando que mais de 90% dos equipamentos sem fio da época já utilizavam conceitos de rádios definidos por software de alguma forma [Forum, 2011]. Isso corrobora com a ideia de que soluções empregando rádios definidos por software são factíveis em uso comercial.

As próximas seções apresentam informações técnicas a respeito da programação de rádios definidos por software, assim como uma visão geral dos principais dispositivos disponíveis no mercado.

## 2.2.1 GNU Radio

GNU Radio é uma plataforma de processamento digital de sinais utilizada na programação de rádios definidos por software [GNURadio, 2018]. A programação é feita através de blocos. A figura 2.6 ilustra o ambiente de desenvolvimento GNU Radio Companion (GRC), parte da plataforma GNU Radio. GRC é dividido em 4 partes [da Silva et al., 2015]: (1) Biblioteca de blocos instalados no GRC; (2) Barra de tarefas com algumas funcionalidades do ambiente de desenvolvimento; (3) Terminal com informações e execução; e (4) *Workspace*, onde o projeto é montado.

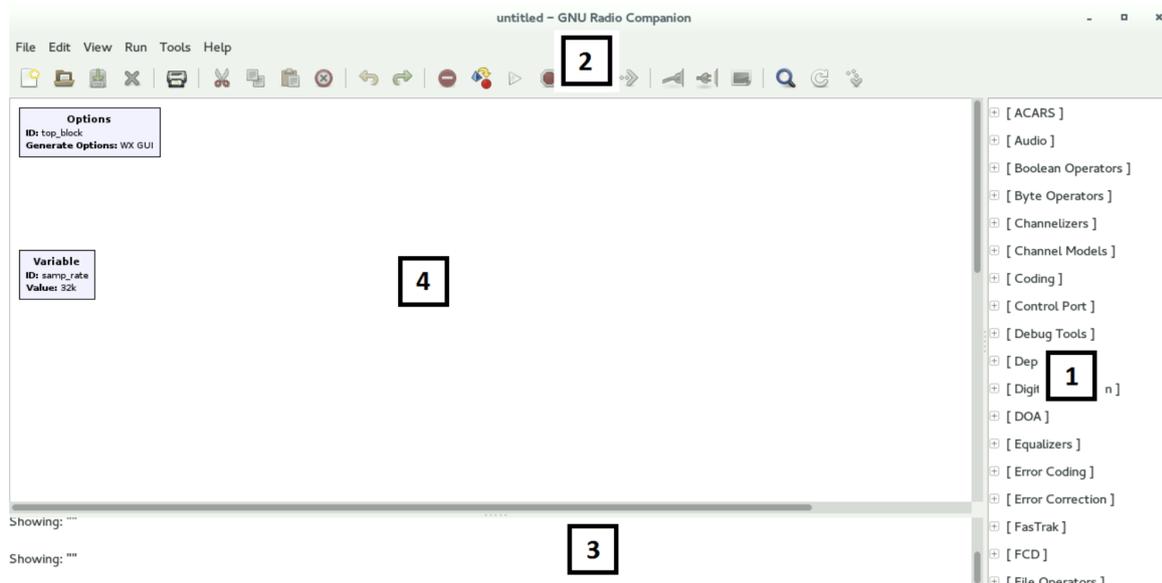


Figura 2.6: GNU Radio Companion (GRC), [da Silva et al., 2015]

A programação é feita através da seleção de blocos a partir da Biblioteca para o *Workspace*. Os blocos são então interligados formando um diagrama de fluxo. Dis-

positivos SDR possuem blocos específicos de acordo com o equipamento. *Drivers* são usualmente fornecidos pelos fabricantes. Esses blocos podem ser do tipo *source* ou *sink*. Blocos *source* fornecem dados para a plataforma. Sendo assim, um bloco *source* de um rádio SDR atua como o canal de recepção de dados (Rx). Analogamente, blocos *sink* funcionam de maneira inversa, recebendo dados da plataforma e atuando como o canal de transmissão de dados (Tx).

Novos blocos podem ser programados. As linguagens de programação suportadas são C++ e Python [GNURadio, 2017]. A escolha da linguagem de programação depende da aplicação. [GNURadio, 2017] ressalta que aplicações de processamento de sinais e/ou em tempo real devem ser feitas em C++ para melhor desempenho. Por outro lado, Python permite maior facilidade e rapidez e pode ser uma boa alternativa em situações nas quais o processamento em tempo real não é importante.

### 2.2.2 Dispositivos SDR

Há diversas opções de rádios definidos por software disponíveis no mercado atualmente. As diferenças geralmente são relacionadas à frequência de operação e/ou hardware disponível. Abaixo, uma breve descrição de plataformas SDR populares e suas respectivas características.

- **USRP** (*Universal Software Radio Peripheral*): Trata-se de uma plataforma de desenvolvimento SDR desenvolvida pela *Ettus Research*<sup>3</sup> com suporte ao GNU Radio. A família de produtos USRP conta com soluções de radiofrequência variando de alguns KHz a 6 GHz, a depender do modelo. A transferência de dados é usualmente feita por USB (*Bus series*), rede (*Network series*) ou PCIe (*X series*). Modelos populares são: B200, B210, N200, N210, X300 e X310. A primeira letra de cada modelo representa o tipo de conexão disponível (B: *Bus series*, N: *Network series* e X: PCIe). Já a numeração indica o modelo. Os modelos geralmente se diferem por frequência de operação, taxa de amostragem e capacidade da FPGA (sigla do inglês *Field Programmable Gate Array*) [Ettus Research, 2018a].
- **HackRF One**: Uma plataforma SDR *open source* desenvolvida e mantida por [Great Scott Gadgets, 2018]. Originalmente, o projeto HackRF One opera com frequências de 1 MHz a 6 GHz, taxa de amostragem de 20 milhões amostras por segundo e possui suporte ao GNU Radio. Contudo, essas especificações podem variar devido à natureza *open source* do projeto (hardware e software), permitindo customizações.

---

<sup>3</sup><https://www.ettus.com/>

- **Sora:** Trata-se de uma plataforma SDR desenvolvida por [Tan et al., 2011]. A proposta combina o poder de processamento de FPGAs com a facilidade de programação de arquiteturas GPPs (*General-Purpose Processor*). A interface de comunicação é PCIe-x8. Por outro lado, Sora não possui rádios de comunicação nativos, necessitando que módulos extras compatíveis sejam adicionados [The Sora Core Team, 2012].
- **RTL-SDR:** SDR de baixo custo capaz de operar em frequências de 500 KHz a 1.75 GHz baseado no *chipset* RTL2832U, originalmente utilizado como receptor DVB-T TV (sigla do inglês *Digital Video Broadcasting - Terrestrial*) [RTL-SDR, 2018]. A taxa de amostragem é usualmente limitada a 3 milhões de amostras por segundo, mas pode variar de acordo com a versão. RTL-SDR é limitado apenas à recepção de sinais, não sendo capaz de transmitir.

A escolha do SDR deve ser feita levando-se em consideração restrições técnicas como frequências de RF suportadas e taxa de amostragem. Contudo, o custo também é um fator limitante, já que dispositivos mais sofisticados como rádios USRP possuem custo elevado<sup>4</sup>.

*Testbeds* compartilhados são uma alternativa ao alto custo. Iniciativas como [FUTEBOL, 2018] disponibilizam rádios definidos por software para acesso remoto sem custo ao usuário. Particularmente, [FUTEBOL, 2018] disponibiliza rádios USRP em três diferentes *testbeds*, sendo eles:

- **Universidade Federal de Minas Gerais (UFMG):** 8 rádios USRP disponíveis entre modelos B200 e B210<sup>5</sup>;
- **Trinity College Dublin (TCD):** 18 rádios USRP N210<sup>6</sup>;
- **Universidade Federal do Rio Grande do Sul (UFRGS):** 6 rádios USRP NI-2901<sup>7</sup>.

A validação da proposta desta dissertação é feita utilizando-se recursos do projeto [FUTEBOL, 2018]. Seções posteriores detalham a utilização desses recursos, assim como a arquitetura e implementação da proposta em rádios definidos por software.

---

<sup>4</sup>USRP B210: R\$6.735.00/unidade, orçamento feito em 08/10/2018 no site <https://www.ettus.com/product/category/USRP-Bus-Series>.

<sup>5</sup><http://futebol.dcc.ufmg.br/equipaments.html>

<sup>6</sup><https://iris-testbed.connectcentre.ie/>

<sup>7</sup><http://futebol.inf.ufrgs.br/index.php/equipment-list/>

## 2.3 Aprendizado por Reforço

Aprendizado por reforço é uma técnica de aprendizado de máquina que consiste no aprendizado de ações através de tentativas e erros [Sutton and Barto, 1998]. O processo é guiado por recompensas e punições (i.e. recompensas negativas). O objetivo é aprender um conjunto ou sequência de ações que maximize as recompensas recebidas. Contudo, essa sequência de ações não é informada a priori. O sistema deve descobri-la ao testar certas ações e ser recompensado ou punido por tê-las escolhido.

Aprendizado por reforço se difere de outras formas de aprendizado de máquina como aprendizado supervisionado e aprendizado não-supervisionado [Sutton and Barto, 1998]. Aprendizado supervisionado consiste no aprendizado a partir de um conjunto de treino rotulado preparado previamente. O sistema aprende a mapear entradas e saídas através desse conjunto e extrapola o conhecimento obtido para situações fora do conjunto de treino. Aprendizado não-supervisionado é tipicamente denominado como "o processo de encontrar de estruturas ocultas de um conjunto de dados não rotulados" [Sutton and Barto, 1998] (tradução livre). Algoritmos de clusterização são exemplos de técnicas de aprendizado não-supervisionado, onde classes são formadas a partir do agrupamento dos dados de características semelhantes. Já aprendizado por reforço consiste no processo de tomada de ações que maximizem um dado sistema de recompensas.

A figura 2.7 ilustra o processo de aprendizado por reforço. O agente é responsável pela tomada de ações. A escolha das ações é feita de tal forma a maximizar a recompensa. Os estados representam as possíveis condições do sistema. Sendo assim, o agente está no estado  $S_t$  e toma a ação  $A_t$  no tempo  $t$ . Essa ação pode modificar o ambiente e levar o agente ao estado  $S_{t+1}$ . Consequentemente, a ação  $A_t$  gera uma recompensa  $R_{t+1}$  (ou punição caso  $R_{t+1} < 0$ ). A recompensa  $R_{t+1}$  diz ao agente o quão bom ou ruim foi a escolha da ação  $A_t$  quando o agente estava no estado  $S_t$ . Assim, o agente pode repetir ou evitar essa ação no futuro com o intuito de maximizar a recompensa a ser recebida.

Um exemplo de aprendizado por reforço é a coleta automatizada de lixo [Sutton and Barto, 1998]. Um robô é responsável por visitar um conjunto de salas e procurar por lixo. O objetivo é coletar a maior quantidade de lixo possível. Por outro lado, o robô é alimentado por baterias e deve sempre manter-se operante. Nesse cenário, o robô deve decidir se entra em uma nova sala à procura de lixo ou se vai até a estação de recarga, dado a facilidade de encontrar uma estação (baseado em experiências prévias) e a carga disponível nas baterias. Em relação à figura 2.7, o robô é o agente e toma ações como: a) procurar por lixo ou b) procurar por uma estação de recarga. O ambiente é

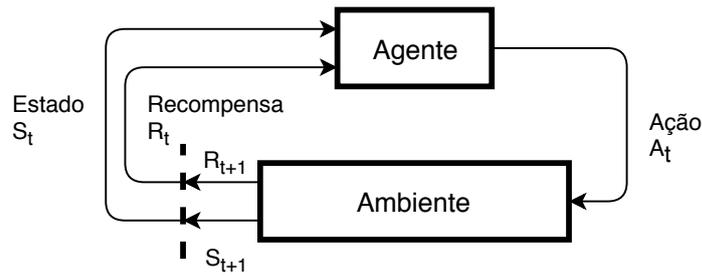


Figura 2.7: Visão geral de um sistema de aprendizado por reforço, adaptado de [Sutton and Barto, 1998]

o mapa de distribuição das salas, lixo e estações de recarga. Os estados são descritos pela carga disponível nas baterias e a facilidade de encontrar uma estação de recarga. Recompensas são obtidas a partir da coleta de lixo; o robô é punido ( $R_t < 0$ ) caso fique sem bateria.

O agente deve tentar diferentes combinações de ações e estados ( $A_t, S_t$ ) para tomar conhecimento das ações ótimas. Esse processo é denominado exploração. Se por um lado o processo de exploração permite o aprendizado do conjunto de ações ótimas, por outro, a exploração pode ter um alto custo em termos de recompensas. A tomada de ações não ou pouco conhecidas pode levar a menores recompensas ou até mesmo punições. Assim, é preferível confiar no conhecimento já adquirido (i.e. explorar). Contudo, essa estratégia também é falha, uma vez que o conhecimento obtido pode ser limitado ou o ambiente (figura 2.7) pode sofrer alterações. Esse dilema entre explorar e confiar no conhecimento já obtido é denominado exploração e exploração (do inglês *exploration vs. exploitation dilemma*). O balanceamento entre explorar e explorar é um dos principais desafios da modelagem de sistemas de aprendizado por reforço [Sutton and Barto, 1998].

Embora os conceitos básicos se preservem, há diferentes técnicas de aprendizado por reforço. A próxima seção apresenta Q-Learning, um dos principais marcos de aprendizado por reforço [Sutton and Barto, 1998].

### 2.3.1 Q-Learning

Q-Learning é uma técnica introduzida por [Watkins, 1989] e é considerada um marco em aprendizado por reforço [Sutton and Barto, 1998]. A qualidade de uma ação  $A_t$  dado um estado  $S_t$  é descrita por  $Q(S_t, A_t)$ . A equação 2.1 define o processo de atualização dos valores de qualidade.  $R_{t+1}$  é o valor da recompensa por ter tomado a ação  $A_t$ . Os parâmetros  $\alpha$  e  $\gamma$  influenciam no quão rápido Q-Learning aprende e no valor de uma possível ação posterior, respectivamente.

$$Q(S_t, A_t) = (1 - \alpha)Q(S_t, A_t) + \alpha[R_{t+1} + \gamma \max_a Q(S_{t+1}, a)] \quad (2.1)$$

Q-Learning pertence à classe de algoritmos de aprendizado por reforço independentes de modelo (do inglês *model-free*) [Sutton and Barto, 1998]. Nessa classe, o agente desconhece como o ambiente se comportará em resposta a uma ação tomada. O comportamento do agente será modelado considerando-se apenas a recompensa consequente da ação, não sendo necessário saber o porquê ou como daquela recompensa. Portanto, técnicas independentes de modelo como o Q-Learning aprendem através do processo de tentativa e erro.

A escolha da ação  $A_t$  é usualmente feita escolhendo-se a ação de maior qualidade dado o estado atual. Contudo, a visão do agente sobre a qualidade de cada ação pode ser limitada ou estar desatualizada. Nesse ponto, o agente enfrenta o dilema de exploração e exploração.

Uma das formas de garantir a exploração é a seleção de ações aleatórias com probabilidade  $\epsilon$ , onde  $0 \leq \epsilon \leq 1$  [Tijmsma et al., 2016]. Essa abordagem é conhecida como  $\epsilon$ -greedy e funciona de acordo com o algoritmo 1, onde *argrandom*  $Q(S, a)$  representa a seleção uniformemente aleatória de uma ação  $a$  dado o estado  $S$ . A característica exploratória do algoritmo cresce com  $\epsilon$ . Quando  $\epsilon = 1$ , a seleção de ações é completamente aleatória;  $\epsilon = 0$ , ações são selecionadas exclusivamente de acordo com  $Q(S_t, A_t)$ . Um ponto negativo dessa estratégia é que ações são selecionadas de forma uniformemente aleatória durante a exploração. Isso pode resultar em um processo de exploração ineficiente, selecionando ações ruins em sequência [Sutton and Barto, 1998].

---

**Algoritmo 1** Q-Learning com estratégia de exploração  $\epsilon$ -greedy
 

---

**Entrada**  $\alpha, \gamma, \epsilon$

Inicializar  $Q(s, a), \forall s \in S, \forall a \in A(s)$

Inicializar  $S_t$

**repita**

**se** *random*  $< \epsilon$  **então**

$A_t \leftarrow \underset{a}{\text{argrandom}} Q(S_t, a)$

$\triangleright$  Exploração  $\epsilon$ -greedy

**senão**

$A_t \leftarrow \underset{a}{\text{argmax}} Q(S_t, a)$

**fim se**

  Tomar ação  $A_t$ , esperar por  $R_{t+1}, S_{t+1}$

$Q(S_t, A_t) \leftarrow (1 - \alpha)Q(S_t, A_t) + \alpha[R_{t+1} + \gamma \max_a Q(S_{t+1}, a)]$

$t \leftarrow t + 1$

**fim repita**

---

---

**Algoritmo 2** Q-Learning com estratégia de exploração softmax

---

**Entrada**  $\alpha, \gamma, T$ Inicializar  $Q(s, a), \forall s \in S, \forall a \in A(s)$ Inicializar  $S_t$ **repita** $A_t \leftarrow \underset{a}{\text{argrandom}}Q(S_t, a)$  de acordo com  $\pi(S_t, a), \forall a \in A(S_t)$  ▷ SoftmaxTomar ação  $A_t$ , esperar por  $R_{t+1}, S_{t+1}$  $Q(S_t, A_t) \leftarrow (1 - \alpha)Q(S_t, A_t) + \alpha[R_{t+1} + \gamma \underset{a}{\text{max}}Q(S_{t+1}, a)]$  $\pi(S_t, a) = \frac{e^{Q(S_t, a)/T}}{\sum_{i=1}^m e^{Q(S_t, a_i)/T}}, \forall a \in A(S_t)$  ▷ Atualização distribuição Boltzmann $t \leftarrow t + 1$ **fim repita**

---

---

**Algoritmo 3** Q-Learning com estratégia de exploração UCB

---

**Entrada**  $\alpha, \gamma, C$ Inicializar  $Q(s, a), \forall s \in S, \forall a \in A(s)$ Inicializar  $\text{bonus}(s, a) \leftarrow 0, \forall s \in S, a \in A(s)$ Inicializar  $S_t$ **repita** $A_t \leftarrow \underset{a}{\text{argmax}}[\text{bonus}(S_t, a) + Q(S_t, a)]$  ▷ Exploração com UCBTomar ação  $A_t$ , esperar por  $R_{t+1}, S_{t+1}$  $Q(S_t, A_t) \leftarrow (1 - \alpha)Q(S_t, A_t) + \alpha[R_{t+1} + \gamma \underset{a}{\text{max}}Q(S_{t+1}, a)]$  $\text{bonus}(S_t, a) = c \times \sqrt{\frac{\log N(S_t)}{N(S_t, a)}}, \forall a \in A(S_t)$  $t \leftarrow t + 1$ **fim repita**

---

Outra estratégia de exploração é a softmax (ou Boltzmann). Softmax atribui uma probabilidade a cada par  $(S_t, A_t)$ . A exploração é direcionada, privilegiando ações com maior probabilidade de otimalidade. A função de probabilidade para escolha de ações é dada pela equação 2.2. O parâmetro  $T$  é denominado temperatura e define características da distribuição de Boltzmann (equação 2.2) assim como a natureza exploratória dessa estratégia [Tijmsa et al., 2016]. Quando  $T \rightarrow 0$ , o comportamento exploratório é reprimido; quando  $T \rightarrow \infty$ , a escolha das ações segue uma distribuição uniforme. O algoritmo 2 descreve o funcionamento do algoritmo Q-Learning com a estratégia de exploração softmax, onde a escolha das ações é aleatória de acordo com a distribuição de probabilidades  $\pi(S_t, A_t)$ .

$$\pi(S_t, A) = \frac{e^{Q(S_t, A)/T}}{\sum_{i=1}^m e^{Q(S_t, a_i)/T}} \quad (2.2)$$

UCB (*Upper-Confidence-Bound*) também é uma alternativa de estratégia de ex-

ploração. Ela é baseada no problema *multi-armed bandit*<sup>8</sup>, e possui versões adaptadas para o algoritmo Q-Learning [Tijmsma et al., 2016]. A característica exploratória é dada por um bônus de não seleção de ações. A função de bônus é dada pela equação 2.3, onde  $N(S_t)$  denota quantas vezes o estado  $S_t$  foi visitado e  $N(S_t, A_t)$  um contador de quantas vezes a ação  $A_t$  foi tomada a partir do estado  $S_t$ . O fator de bônus representa a incerteza relacionada a ações pouco selecionadas, forçando a seleção ao longo do tempo. O parâmetro  $c$  controla o grau de exploração [Sutton and Barto, 1998]. Q-Learning com estratégia de exploração UCB é descrito pelo algoritmo 3, onde a ação é selecionada com o intuito de maximizar a qualidade  $Q(S_t, A_t)$  somada à função  $bonus(S_t, A_t)$ .

$$bonus(S_t, A_t) = c \times \sqrt{\frac{\log N(S_t)}{N(S_t, A_t)}} \quad (2.3)$$

### 2.3.2 Q-Learning em ambientes não estacionários

O ambiente sobre o qual o agente toma ações pode sofrer alterações externas ao longo do tempo. Nesse caso, o ambiente é dito não estacionário, e a qualidade das ações pode ser afetada por essas mudanças externas [Sutton and Barto, 1998]. Um exemplo de ambiente não estacionário é o robô coletor de lixo (exemplo da seção 2.3), onde a localização da estação de recarga muda ao longo do tempo. Nesse caso, a qualidade de cada ação (entrar na sala e procurar por lixo **ou** procurar a estação de recarga) é afetada pelo quão fácil é encontrar a estação de recarga dada a nova localização.

Apesar de acrescentar uma dificuldade extra, ambientes não estacionários são comuns em grande parte dos problemas de aprendizado por reforço [Sutton and Barto, 1998]. Nesses casos, abordagens exploratórias são necessárias para assegurar que as ações de maior qualidade ou com maiores recompensas sejam atualizadas e, posteriormente, escolhidas ao longo do tempo. Além disso, [Sutton and Barto, 1998] cita que métodos de atualização simples como o da equação 2.4 podem ser utilizados em ambientes não estacionários, onde  $\alpha$  representa a taxa de atualização e  $G_t$  o ganho atual no tempo  $t$ .

$$V(S_t) \leftarrow V(S_t) + \alpha[G_t - V(S_t)] \quad (2.4)$$

Q-Learning lida com ambientes não estacionários. A equação de atualização do algoritmo Q-Learning (equação 2.1) corrobora com a equação 2.4 de atualização em

---

<sup>8</sup>Nome inspirado no problema de seleção de alavancas durante um jogo contra múltiplas máquinas caça-níqueis. A generalização consiste em escolher uma ação entre  $k$  ações possíveis e, posteriormente, receber uma recompensa baseada na ação escolhida [Sutton and Barto, 1998].

ambientes não estacionários, sendo ambos os parâmetros  $\alpha$  relacionados à taxa de atualização (ou aprendizagem) e  $G_t$  equivalente à recompensa  $R_t$ . A atualização da função  $Q(S_t, A_t)$  pode ser feita de maneira *online*, onde o *loop* interno dos algoritmos 1, 2 e 3 representa a execução do algoritmo ao longo do tempo. Adicionalmente, as variações  $\epsilon$ -greedy, softmax e UCB (seção 2.3.1) introduzem a abordagem exploratória necessária em ambientes não estacionários [Sutton and Barto, 1998], garantindo a convergência do algoritmo.

## 2.4 Resumo

Esse capítulo apresentou os conceitos base necessários para o entendimento da proposta desta dissertação. As primeiras seções abordam as funcionalidades da subcamada MAC, assim como uma breve descrição dos protocolos MAC CSMA/CA e TDMA. Discorreremos também que protocolos MAC tendem a ser ótimos em configurações específicas de rede, embora redes sem fio sejam usualmente dinâmicas. Depois, foi dada uma visão geral dos rádios definidos por software, programabilidade e principais dispositivos disponíveis no mercado. Por último, apresentamos conceitos de aprendizado por reforço, com foco particular no algoritmo Q-Learning.

Os próximos capítulos desta dissertação juntam os conceitos apresentados aqui. Trata-se do chaveamento de protocolos MAC de acordo com o estado da rede sem fio através de aprendizado por reforço, com implementação feita em rádios definidos por software.

# Capítulo 3

## Trabalhos relacionados

Este capítulo resume os principais trabalhos encontrados na literatura relacionados ao tema desta dissertação. A metodologia de pesquisa inclui, principalmente, buscas em plataformas digitais como Google Scholar<sup>1</sup> e IEEE Xplore Digital Library<sup>2</sup>. Algumas palavras chave utilizadas no processo de busca são: *hybrid MAC protocol*, *MAC protocol selection* e *adaptive MAC protocol*.

O capítulo é dividido em três seções. A primeira apresenta trabalhos baseados em protocolos MAC híbridos. A segunda aborda trabalhos que utilizam técnicas de adaptação de protocolos MAC. Por fim, a última seção descreve trabalhos relacionados à troca de protocolos MAC, sendo essa a categoria da proposta desta dissertação (SOMAC – Subcamada auto-Organizável MAC).

### 3.1 Protocolos MAC híbridos

Protocolos MAC híbridos juntam características de dois ou mais protocolos. O objetivo é obter o melhor dos mundos. Contudo, essa abordagem muitas vezes também resulta em uma junção das desvantagens de cada protocolo ou culmina em novas limitações em termos de cenários ótimos de uso.

[Rhee et al., 2008] apresenta um protocolo MAC híbrido para redes de sensores sem fio denominado Z-MAC. A proposta é uma mistura dos protocolos CSMA e TDMA. O acesso ao meio é dividido em *slots* de tempo similarmente ao TDMA, onde cada nó é dono de um *slot*. Os nós da rede disputam o uso de cada *slot* de tempo de maneira similar ao CSMA, primeiro escutando o meio sem fio e, caso o meio esteja disponível, iniciando a transmissão. Em caso de disputa com o dono do *slot*, esse possui prioridade

---

<sup>1</sup><https://scholar.google.com>

<sup>2</sup><https://ieeexplore.ieee.org>

para transmissão. Assim, *slots* não ficam ociosos em momentos de baixa contenção (vantagem CSMA) e há uma hierarquia de transmissão em momentos de alta disputa (vantagem TDMA). Todavia, Z-MAC possui inicialização custosa e é dependente da topologia da rede. Conseqüentemente, Z-MAC também sofre com a mobilidade dos nós sem fio dentro da rede, sendo necessário reiniciar o processo de inicialização e descoberta da topologia.

LA-MAC (*Load-Adaptive MAC*) é um protocolo MAC híbrido proposto por [Hu et al., 2011]. Há dois modos de operação, CSMA e HYBRID. O modo HYBRID mistura conceitos dos protocolos CSMA e TDMA e possui dois estados de operação: a) baixa contenção e b) alta contenção. A contenção é estimada através da taxa de colisão. Quando a contenção é baixa, todos os nós podem competir por qualquer *slot* de tempo de transmissão no mesmo estilo do protocolo CSMA. Similarmente ao Z-MAC, há um esquema de prioridades em caso de disputa. LA-MAC lida com prioridades alterando o tamanho da janela de contenção, sendo maior a prioridade, menor o tamanho da janela. Já em situações de alta contenção, a disputa por *slots* de tempo é feita somente entre o dono do *slot* e vizinhos próximos, limitando o número de concorrentes. Novamente, o esquema de prioridades é baseado no tamanho da janela de contenção. Como desvantagem, LA-MAC depende de informações sobre a topologia da rede quando operando sob alta contenção e, portanto, é sensível à mobilidade de nós sem fio.

O trabalho de [Yang et al., 2018], chamado CTh-MAC, também é um híbrido entre os protocolos CSMA e TDMA. A proposta assume comunicação todos para um, onde toda a comunicação da rede é feita através de um nó denominado *sink*. CTh-MAC divide a rede em subconjuntos. A divisão é baseada na distância dos nós sem fio com o nó *sink*, calculada através de atrasos de comunicação entre o envio de mensagens pelo nó *sink* e o recebimento das respectivas respostas. Cada subconjunto recebe um *slot* de tempo, de forma análoga ao protocolo TDMA. Por sua vez, os nós sem fio competem dentro do *slot* de tempo de seu subconjunto de forma análoga ao protocolo CSMA. Como resultado, [Yang et al., 2018] indica melhoras em termos de vazão de dados e consumo de energia. Todavia, a principal desvantagem dessa abordagem está na dependência da topologia todos para um. Outro ponto desfavorável é a necessidade periódica de refazer o cálculo dos subconjuntos para tratar a mobilidade de nós.

Protocolos híbridos diferem da proposta desta dissertação, uma vez que consistem na criação de novos protocolos MAC, enquanto SOMAC consiste em um mecanismo de seleção de protocolos. Esses protocolos podem, inclusive, fazer parte da subcamada MAC de SOMAC, que os selecionaria quando a rede sem fio apresente condições favoráveis de uso.

## 3.2 Protocolos MAC adaptáveis

Esta seção foca em protocolos MAC adaptáveis. A característica adaptável pode ser tanto resultado de ajuste de parâmetros de protocolos MAC já existentes, como também a criação de protocolos genéricos capazes de se adaptarem ao funcionamento de uma rede sem fio em operação.

O trabalho de [Libin Jiang and Walrand, 2010] consiste em um protocolo CSMA adaptável. A proposta é modelada usando-se cadeias de Markov, onde os padrões de tráfego são utilizados no cálculo das probabilidades de transição. Essas, por sua vez, ditam a taxa de agressividade de cada nó, diretamente relacionada à respectiva janela de contenção. Quanto mais agressivo, menor a janela de contenção do nó sem fio. Embora [Libin Jiang and Walrand, 2010] aponte resultados positivos derivados de simulações computacionais, algumas ressalvas devem ser levadas em consideração. A primeira diz respeito à suposição de um ambiente perfeito, onde não há ocorrência dos problemas de nó escondido e nó exposto. Ou seja, alguns problemas intrínsecos do protocolo CSMA são preservados. Segundo, [Libin Jiang and Walrand, 2010] assume ambientes estacionários ou quase estacionários (mudanças lentas). Para satisfazer essa premissa, pacotes postiços (*dummy packets*) são utilizados para manter o tráfego médio da rede quase constante. Isso não apenas promove a ocupação desnecessária do meio sem fio, como também pode não representar a realidade de redes sem fio reais, onde padrões de tráfego podem mudar múltiplas vezes ao longo do tempo.

[Yu et al., 2017] propõe a criação de um protocolo MAC universal através de técnicas de aprendizado por reforço profundo (do inglês *Deep-Reinforcement Learning*) denominado DLMA (*Deep-reinforcement Learning Multiple Access*). A proposta é focada na utilização de *slots* de tempo ociosos quando interoperando com protocolos baseados na alocação de *slots*. Especificamente, DLMA é testado contra os protocolos TDMA e *slotted* ALOHA. Cenários de avaliação simples como um nó com DLMA e outro com TDMA ou ALOHA apontam bons resultados. Todavia, [Yu et al., 2017] carece de cenários de avaliação mais complexos. Múltiplos agentes de aprendizado por reforço tomando decisões sobre o mesmo ambiente resultam, por si só, em não estacionariedade, dificultando a convergência do sistema [Marinescu, 2016]. [Yu et al., 2017] apresenta avaliações para cenários com 3 nós DLMA, 1 TDMA e 2 *slotted* ALOHA. Contudo, não é apresentado o impacto dos nós DLMA sobre os demais em termos de desempenho. Adicionalmente, o cenário é consideravelmente determinístico, onde o nó TDMA sempre utiliza apenas 3/10 dos *slots* disponíveis e os nós *slotted* ALOHA transmitem com um probabilidade fixa de 20%, representando pouco a variabilidade de cenários reais.

Protocolos adaptáveis são, por si só, protocolos MAC. Similarmente aos protocolos híbridos, protocolos adaptáveis podem ser adicionados à subcamada MAC de SOMAC, sendo selecionados quando o estado da rede sem fio apresentar condições favoráveis de uso.

### 3.3 Troca de protocolos MAC

Ao contrário das seções anteriores, os trabalhos listados aqui não apresentam modificações de protocolos MAC *per se*. Otimizações da subcamada MAC são propostas através de um sistema de troca de protocolos. A proposta desta dissertação se encaixa nessa categoria. Assim, os trabalhos abaixo são os de maior correlação com SOMAC dentre os apresentados neste capítulo.

AMAC (*Adaptive MAC*) possibilita a troca de protocolos MAC, sendo os protocolos CSMA e TDMA utilizados na avaliação de [Huang et al., 2009]. A avaliação é experimental e é baseada em rádios definidos por software. [Huang et al., 2009] propõe dois métodos de seleção de protocolos. Primeiro, a troca é engatilhada sempre que há degradação de desempenho além de um dado limiar, sendo 20% o valor utilizado em AMAC. Esse método é rudimentar em situações nas quais a degradação afeta ambos os protocolos como, por exemplo, na presença de um agente externo gerador de ruído ou uma queda na demanda de tráfego na rede. Nesses casos, a troca de protocolos pode não ser uma boa ação. A segunda estratégia é baseada na predição de padrões de tráfego futuros. Especificamente, AMAC prediz o tamanho médio dos pacotes a serem transmitidos pela rede. Caso o tamanho médio seja dito grande, a rede tenderá a operar com o protocolo TDMA; caso contrário, o protocolo CSMA será utilizado. Todavia, o desempenho dos protocolos MAC não é limitado ao tamanho dos pacotes. Frequência, potência, modulação, mobilidade etc. podem afetar no desempenho da rede como mencionado pelo próprio autor, podendo cada protocolo ser afetado de maneira diferente.

Já [Ye et al., 2016] utiliza de modelagens matemáticas para o processo de seleção de protocolos MAC. A solução consiste na troca entre os protocolos DCF (*Distributed Coordination Function*) e D-TDMA (*Dynamic Time Division Multiple Access*). O ponto chave do trabalho é a definição de *switching points*, quando a troca de protocolos se faz necessária. Para tal, modelos matemáticos para cada protocolo são propostos para situações onde a rede está saturada e não saturada. Assim, os *switching points* são obtidos e testados via simulações. Essa abordagem é custosa em termos de escalabilidade. A inserção de um novo protocolo MAC necessita da modelagem matemática de

um especialista para a obtenção dos novos *switching points*. Além disso, a modelagem proposta pouco considera características do meio, considerando, inclusive, um canal sem fio livre de erros durante as simulações. Essa abordagem pode não representar as características de uma rede sem fio real, o que impacta na generalização dos resultados obtidos.

O trabalho de [Cordeiro et al., 2018] consiste em uma plataforma flexível da subcamada MAC denominada FS-MAC (*Flexible System for Medium Access Control*). A plataforma é implementada em rádios definidos por software e permite a troca de protocolos MAC com a rede sem fio em operação. Os protocolos disponíveis são CSMA e TDMA, e a implementação é baseada no padrão IEEE 802.15.4. Embora FS-MAC seja a plataforma base desta dissertação, é importante destacar as principais diferenças. Primeiro, a implementação de SOMAC é baseada no padrão IEEE 802.11, trazendo resultados mais próximos às redes sem fio populares. Segundo e mais importante, o módulo de seleção de protocolos MAC de FS-MAC é baseado em lógica *fuzzy*. Ainda que resultados positivos possam ser extraídos, essa abordagem é dependente da configuração manual de regras de inferência. Essas regras devem ser definidas por um especialista considerando-se tanto a configuração da rede sem fio quanto características de cada protocolo MAC. SOMAC, por outro lado, possui um módulo de seleção de protocolos MAC autônomo, sendo capaz de adaptar-se às mudanças da rede sem a necessidade de um especialista.

[Qiao et al., 2016] apresenta um mecanismo de seleção de protocolos MAC baseado em técnicas de aprendizado de máquina (AM), ao qual nos referimos como SMAC (*Switching MAC protocols*). Dois protocolos MAC estão disponíveis, CSMA e TDMA. A seleção de protocolos é feita via um classificador baseado em técnicas de aprendizado supervisionado, sendo SMO (*Sequential Minimal Optimization*) o algoritmo utilizado. A natureza de aprendizado supervisionado exige que um conjunto de treino rotulado seja previamente fornecido. A construção desse conjunto é feita via simulações computacionais. [Qiao et al., 2016] repetem cenários de simulação de redes para ambos os protocolos. Assim, é possível comparar o desempenho de cada um e rotular amostras de parâmetros de rede com o respectivo protocolo ótimo. Dois problemas surgem dessa abordagem. Primeiro, a construção do conjunto de treino é custosa em ambientes reais. A repetição fiel de cenários de redes para ambos os protocolos pode ser inviável ou até mesmo impossível. Segundo, o algoritmo é limitado ao conjunto de treino. Em ambientes não estacionários, a resposta da rede pode mudar ao longo do tempo. Isso pode resultar em situações nas quais o conjunto de treino não mais represente o comportamento da rede, comprometendo a acurácia do classificador de SMAC.

Similarmente, [Gonçalves et al., 2018] apresenta um mecanismo de seleção de

protocolos MAC baseado em aprendizado de máquina supervisionado denominado FS-MAC+, contando com os protocolos CSMA e TDMA. Trata-se de uma versão aprimorada do trabalho de [Cordeiro et al., 2018] (FS-MAC). A principal diferença, em relação ao SMAC, está na utilização de regressores para cada um dos protocolos MAC disponíveis ao invés de um classificador. Os regressores estimam o desempenho de cada protocolo periodicamente, sendo o protocolo de melhor desempenho selecionado. Essa abordagem facilita a construção do conjunto de treinamento uma vez que não é mais necessário repetir fielmente estados de rede para ambos os protocolos MAC e, posteriormente, rotulá-los. Contudo, FS-MAC+ não corrobora com ambientes não estacionários como redes sem fio, sendo, assim como SMAC, limitado ao conjunto de treino previamente obtido.

	[Huang et al., 2009]	[Ye et al., 2016]	[Cordeiro et al., 2018]	[Qiao et al., 2016]	[Gonçalves et al., 2018]	SOMAC
Avaliação experimental	✓	-	✓	-	✓	✓
Número de protocolos	2	2	2	2	2	2
Escalabilidade de protocolos	✓	-	✓	-	✓	✓
Independente de especialista	✓	-	-	✓	✓	✓
Ambientes não-estacionários	-	-	-	-	-	✓
Algoritmo de seleção	Li	MM	Fu	AM	AM	AR

Tabela 3.1: Comparação entre trabalhos relacionados (Li - limiar; MM - modelagem matemática; Fu - lógica fuzzy; AM - aprendizado de máquina; AR - aprendizado por reforço)

Em resumo, a tabela 3.1 lista as principais características dos trabalhos descritos nesta seção e os compara com SOMAC. Nela, apenas os trabalhos de [Huang et al., 2009], [Cordeiro et al., 2018], [Gonçalves et al., 2018] e SOMAC possuem avaliação experimental, o que aproxima os resultados obtidos de condições reais de aplicação. O número de protocolos disponíveis é 2 para todos os trabalhos. [Ye et al., 2016] e [Qiao et al., 2016] são de difícil escalabilidade em termos do número de protocolos MAC, requerendo para cada novo protocolo, respectivamente: a) modelagem matemática complexa para obtenção dos *switching points* e b) a construção de um novo conjunto de treino rotulado. [Ye et al., 2016] e [Cordeiro et al., 2018] são dependentes de um especialista para construção (ou ajuste) do mecanismo de seleção. Por fim, [Qiao

et al., 2016], [Gonçalves et al., 2018] e SOMAC contam com técnicas de aprendizado de máquina como método de seleção mas somente SOMAC lida com ambientes não estacionários.

## 3.4 Resumo

Este capítulo lista os principais trabalhos relacionados encontrados na literatura associados a esta dissertação. Embora muitas vezes melhores que os protocolos originais, protocolos MAC híbridos e adaptáveis ainda são, por si só, protocolos MAC e usualmente possuem cenários específicos de uso. A seção 3.3 apresenta os trabalhos baseados em troca do protocolos. Essa é, inclusive, a categoria de SOMAC e, portanto, a mais relevante deste capítulo. Resumidamente, os principais diferenciais de SOMAC em relação aos demais trabalhos são: a) avaliação experimental, b) método de seleção baseado em aprendizado por reforço e c) adaptabilidade a ambientes não estacionários.



# Capítulo 4

## SOMAC: Subcamada auto-Organizável MAC

A solução SOMAC consiste em uma Subcamada auto-Organizável MAC. Protocolos MAC são alternados de acordo com o estado da rede sem fio, sendo o melhor protocolo selecionado para procurar aumentar o desempenho da rede. A seleção é baseada no algoritmo de aprendizado por reforço Q-Learning, trocando-se o protocolo quando necessário.

Este capítulo descreve a arquitetura da solução proposta, apresentando tópicos chave do problema de seleção de protocolos MAC, assim como as abordagens adotadas para solucioná-los. Por fim, detalhes de implementação são apresentados nas seções finais do capítulo.

### 4.1 SOMAC

A seleção do protocolo MAC é centralizada. Nós sem fio periodicamente enviam métricas de rede para uma entidade de decisão. Essa entidade emprega a técnica de aprendizado por reforço Q-Learning para selecionar o melhor protocolo MAC, informando os demais nós da rede sobre a decisão.

#### 4.1.1 Arquitetura de SOMAC

O funcionamento dessa arquitetura pode ser dividido em duas partes. Primeiro, a organização dos nós da rede sem fio. Segundo, os módulos necessários para o funcionamento de SOMAC e as respectivas funcionalidades. As próximas seções discutem cada um desses tópicos separadamente.

#### 4.1.1.1 Organização da rede

O funcionamento da rede é baseado em uma relação mestre-escravo e ilustrado pela figura 4.1. A função de mestre é exercida pelo nó denominado coordenador. Há apenas um nó desse tipo na rede, sendo ele o responsável pela seleção do protocolo MAC. Periodicamente, o nó coordenador envia mensagens de *broadcast* informando qual o protocolo MAC deve ser usado na rede naquele momento, processo referenciado pelos envelopes com mensagens "MAC" da figura 4.1. A periodicidade dessas mensagens é de 5 segundos, valor estabelecido empiricamente.

Os demais nós exercem a função de escravo. Esses nós sem fio operam com o protocolo MAC especificado pelas mensagens de controle recebidas do nó coordenador. Ao conectar-se à rede, um nó escravo não possui protocolo MAC definido e, portanto, não é capaz de transmitir dados. Nesse caso, o nó aguarda a próxima mensagem de *broadcast* do coordenador, passando a operar de acordo com o protocolo especificado a partir de então. Adicionalmente, os nós escravos periodicamente enviam mensagens com métricas de desempenho para o nó coordenador, sendo utilizadas para o processo de decisão de qual protocolo deve estar ativo no momento.

Por conveniência, assumimos uma rede estruturada nesta dissertação, sendo toda a comunicação feita através do nó coordenador, o *gateway* local. Essa estrutura é tipicamente utilizada em redes sem fio, onde os nós comunicam exclusivamente com um ponto de acesso (ou *wireless access point*, WAP). Todavia, a presença de um *gateway* não é obrigatória. A arquitetura também funciona quando nós escravos comunicam diretamente entre si. A única premissa necessária é a de que o nó coordenador possa comunicar-se diretamente com todos os nós da rede para envio e recebimento de mensagens de controle.

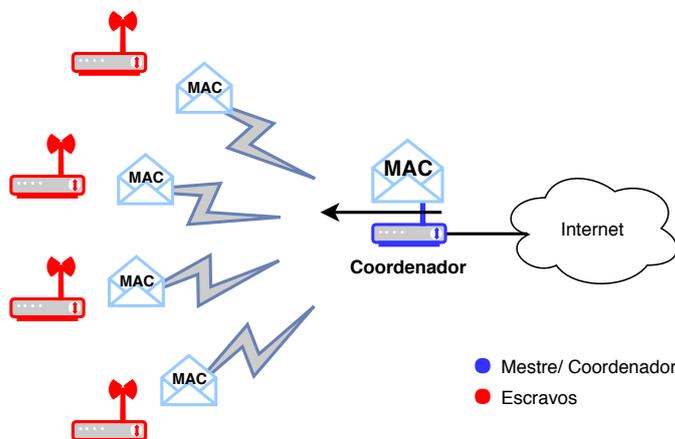


Figura 4.1: Exemplo de topologia utilizada por SOMAC

As métricas de desempenho enviadas pelos nós escravos estão listadas na tabela

4.1. A granularidade de envio das métricas de desempenho é menor que a granularidade da decisão. Dessa forma, agregações são necessárias, sendo possível: soma, média, mínimo, máximo e variância. As agregações são feitas exclusivamente pelo nó coordenador, não sendo transmitidas através da rede. As granularidades utilizadas neste trabalho são 30 e 60 segundos, respectivamente. Esses valores foram estabelecidos empiricamente, sendo pequenos o suficiente para decisões rápidas e grandes o suficiente para robustez diante de variações bruscas de desempenho da rede.

Métrica	Descrição	Agregações
Vazão	Quantidade de quadros por segundo trafegando na rede	soma, média, mín., máx., var.
Latência	Intervalo de tempo entre o envio de um quadro e o recebimento do respectivo ACK	soma, média, mín., máx., var.
Jitter	Desvio das latências ao longo do tempo	soma, média, mín., máx., var.
RNP	<i>Required Number of Packet transmissions</i> , relaciona a quantidade de retransmissões de quadros com a quantidade de transmissões bem sucedidas [Zhou et al., 2015]	soma, média, mín., máx., var.
SNR	<i>Signal-to-Noise Ratio</i> , magnitude do sinal recebido em comparação ao ruído	soma, média, mín., máx., var.

Tabela 4.1: Métricas de desempenho disponíveis para seleção do protocolo MAC

#### 4.1.1.2 Arquitetura de software da subcamada MAC

A arquitetura de software da subcamada MAC é inspirada no trabalho de [Cordeiro et al., 2018]. A figura 4.2 ilustra o esquemático dos módulos de software do SOMAC, podendo ser dividida em três módulos principais: decisão, protocolos MAC e sensoriamento. Uma breve descrição de cada um desses módulos é apresentada abaixo.

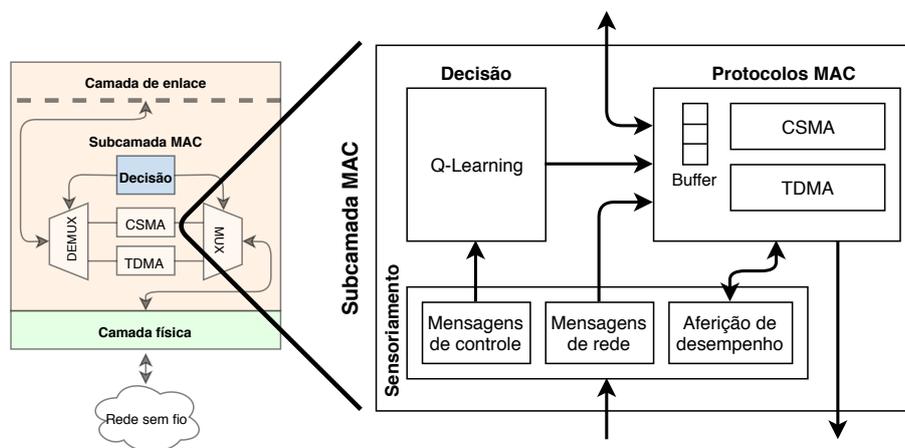


Figura 4.2: Arquitetura de software da subcamada MAC de SOMAC

- **Decisão:** O módulo de decisão seleciona o protocolo MAC a ser utilizado. A seleção do protocolo MAC ocorre apenas no nó coordenador/mestre. As entradas do módulo são as métricas de desempenho reportadas pelos demais nós. O mecanismo de decisão utilizado neste trabalho é a técnica de aprendizado por reforço Q-Learning, como descrito na seção 4.1.2. Após a seleção do protocolo, o módulo de decisão não só altera o protocolo MAC em uso, como também envia mensagens de controle para toda a rede informando sobre a decisão tomada. Nos nós escravos, o módulo de decisão recebe como entrada as mensagens de controle do nó coordenador informando o protocolo MAC e, em sequência, altera o protocolo MAC em uso. Esse processo é ilustrado pela figura 4.3 para uma rede com dois nós escravos. A sequência de ações se repete periodicamente.

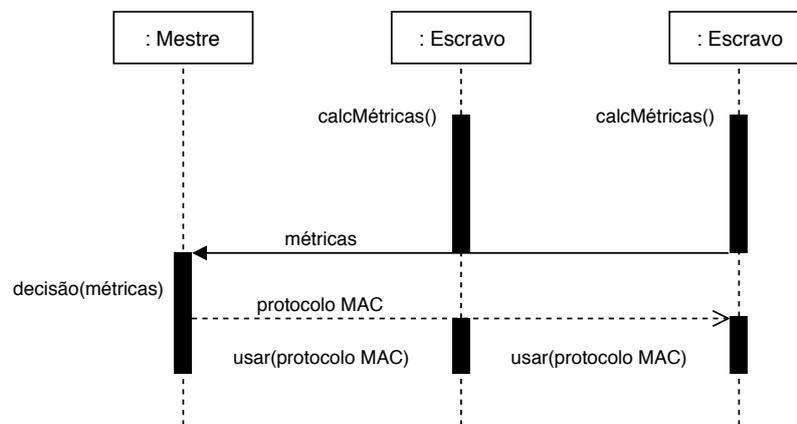


Figura 4.3: Diagrama de decisão

- **Protocolos MAC:** Esse módulo contém os protocolos MAC disponíveis para uso e é controlado pelo módulo de decisão. Há apenas um protocolo MAC ativo por vez e, conseqüentemente, toda a conexão entre a parte superior da camada de enlace e a camada física é feita exclusivamente por esse protocolo, incluindo a identificação de destino do quadro, o tratamento de quadros repetidos e o envio de mensagens de confirmação de recebimento (ACK).
  - **Buffer:** Além disso, um *buffer* independente é mantido para armazenamento de quadros para envio. Uma entidade independente de armazenamento evita que quadros sejam enfileirados localmente por cada protocolo e perdidos caso o protocolo MAC seja alterado.
- **Sensoriamento:** O módulo de sensoriamento classifica e redireciona os quadros recebidos. Mensagens de controle (métricas de desempenho se coordenador; protocolo MAC caso contrário) são encaminhadas para o módulo de decisão. Já

mensagens de rede são redirecionadas para o módulo de protocolos MAC. Adicionalmente, é de responsabilidade do módulo de sensoriamento aferir o desempenho da rede sem fio através do cálculo de métricas de desempenho (tabela 4.1). Periodicamente, o resultado da aferição é enviado para o módulo de protocolos MAC em forma de quadro, cujo endereço destino é o nó coordenador.

Os quadros de controle gerados pelos módulos de decisão e de sensoriamento são distinguidos pelo campo de *frame control* (FC) do cabeçalho MAC do padrão IEEE 802.11 [IEEE, 2016]. FC igual a 0x2900 representa o quadro de controle com o protocolo MAC; Já o valor 0x2100 indica métricas geradas pelo módulo de sensoriamento. Ambos os códigos não fazem parte das especificações de IEEE 802.11 e, portanto, não comprometem o funcionamento de nenhuma outra camada da pilha de protocolos de rede baseada nesse padrão.

Como dito anteriormente, o mecanismo de seleção do módulo de decisão é a principal contribuição desta dissertação. A próxima seção detalha o funcionamento do mecanismos de seleção utilizado em SOMAC.

### 4.1.2 Mecanismo de seleção

A criação de um mecanismo de seleção de protocolos MAC é complexa. Primeiro, redes sem fio são ambientes não estacionários. O desempenho da rede é sensível a variações comportamentais de usuários (e.g. mobilidade e aplicativos utilizados - voz, vídeo ou dados) e variações do meio, sazonais ou não, (e.g. umidade do ar e obstáculos para propagação do sinal). Nesse escopo, técnicas de aprendizado supervisionado resultam em modelos de seleção ineficientes por serem limitadas ao conjunto de treino previamente obtido. Ainda que haja mecanismos de atualização desses modelos ao longo do tempo, eles são focados em fluxos massivos de dados e são propensos a problemas como esquecimento catastrófico (do inglês *catastrophic forgetting*) e estabilidade-plasticidade (do inglês *stability-plasticity*) [Bifet et al., 2010], [Gepperth and Hammer, 2016].

Segundo, modelos para redes sem fio são difíceis de serem construídos. [Ferrand et al., 2016] ressalta que o meio de propagação de redes sem fio é altamente variável no tempo, frequência e espaço. Além disso, características intrínsecas de cada dispositivo tais como ganho da antena, sensibilidade de recepção de sinais e potência de transmissão também afetam no desempenho da rede. Não obstante, o próprio protocolo MAC afeta características da rede sem fio e deve ser considerado pelo modelo. Por exemplo, protocolos baseados em contenção como o CSMA/CA podem gerar maior nível de interferência quando comparados a protocolos sem contenção como o TDMA.

Nesse contexto, técnicas de aprendizado independentes de modelos são preferíveis como mecanismo de seleção de protocolos MAC.

SOMAC utiliza a técnica de aprendizado por reforço Q-Learning. Uma possível alternativa seria a variação de Q-Learning DQN (*Deep Q-Network*), onde redes neurais profundas são utilizadas para a construção da função  $Q$  [Mnih et al., 2015]. Contudo, o ambiente não estacionário de redes sem fio dificulta a utilização de redes neurais, uma vez que o conjunto de treino utilizado pode não representar o comportamento da rede ao longo do tempo. A atualização contínua da rede neural não só adiciona complexidade ao sistema, como também pode afetar no tempo de convergência de DQN [Tsividis et al., 2017]. Assim, a escolha pela forma nativa de Q-Learning não apenas satisfaz os requisitos de não estacionariedade e independência de modelos (capítulo 2, seção 2.3.1) de redes sem fio como também corrobora os resultados de [Sutton and Barto, 1998] e [Kuleshov and Precup, 2014], onde métodos simples tendem a apresentar desempenho igual ou melhor a algoritmos complexos na prática.

Adicionalmente, SOMAC responde a questão tema desta dissertação, reapresentada abaixo. A ação tomada por SOMAC em um instante  $t$  é uma função do protocolo MAC em uso  $P_t$  e da métrica de desempenho  $m_t$ , podendo ser representada como  $P_{t+1} = f(P_t, m_t)$ . A ação é a seleção do próximo protocolo MAC,  $P_{t+1}$ , de tal forma a otimizar a métrica de desempenho  $m$  ao longo do tempo. A decisão, desta forma, pode ser sumarizada como segue.

Dado o atual protocolo MAC e uma métrica de desempenho  $m$ , onde  $m$  pode ser uma única métrica ou um conjunto de métricas de rede agregadas em um valor escalar, deve-se manter ou alterar o protocolo MAC para otimizar a métrica  $m$ ?

As próximas seções detalham a modelagem da técnica Q-Learning no escopo de SOMAC. Especificamente, elas descrevem os pares de estados e ações, o sistema de recompensas e os hiperparâmetros utilizados.

#### 4.1.2.1 Estados e ações

Os estados de Q-Learning são baseados exclusivamente nos protocolos MAC. Cada protocolo é um estado da função  $Q$ . As possíveis ações são manter **ou** alterar o protocolo MAC. Essa abordagem facilita o processo de exploração, uma vez que a dimensionalidade da função  $Q(S_t, A_t)$  é linearmente proporcional apenas ao número de protocolos MAC disponíveis. A simplicidade dessa metodologia vai de encontro ao trabalho de

[Kuleshov and Precup, 2014], sugerindo que heurísticas simples possuem desempenho igual ou melhor a algoritmos complexos na prática.

A definição geral dessa modelagem é formalmente dada pelo processo de decisão de Markov onde o conjunto de estados é definido por  $S = \{P_1, P_2, \dots, P_n\}$  enquanto o conjunto de ações é dado por  $A(s) = \{a_{s1}, a_{s2}, \dots, a_{sn}\}$ . O estado  $P_n$  representa o enésimo protocolo MAC disponível ao passo que  $a_{si}$  representa a ação de mudar para o protocolo  $P_i$  dado o estado  $s$  ou manter o protocolo em uso caso  $s$  seja igual a  $P_i$ . A figura 4.4 ilustra a máquina de estados desse processo de decisão de Markov.

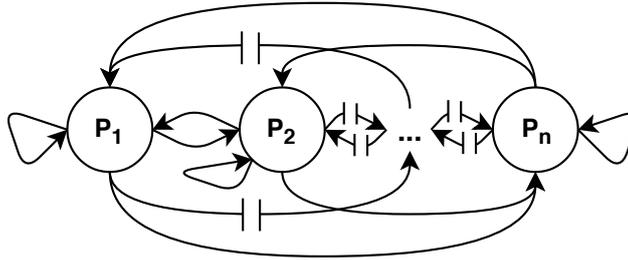


Figura 4.4: Máquina de estados de SOMAC

Apesar de parâmetros da rede sem fio não serem considerados na construção dos estados, SOMAC é sensível a variações da rede. Por exemplo, os estados não distinguem quando o protocolo CSMA/CA está operando sob diferentes níveis de SNR ou quantas estações estão conectadas à rede. Por outro lado, a recompensa obtida por manter ou trocar o protocolo MAC retrata esses e outros parâmetros de rede naquele instante. Sendo assim, SOMAC é sensível a variações da rede através da recompensa obtida ao longo do tempo.

#### 4.1.2.2 Recompensa

O sistema de recompensas é baseado na maximização ou minimização de uma métrica de desempenho  $m$ . Essa métrica pode ser qualquer valor escalar que quantifique o desempenho da rede, podendo ser, inclusive, uma combinação matemática de múltiplas métricas de rede. A tabela 4.1 lista algumas das métricas de rede disponíveis atualmente em SOMAC. Por exemplo, a métrica escolhida pode ser a vazão de dados na rede. Caso ocorra uma troca de protocolos e um aumento de vazão, o sistema recebe uma recompensa positiva ( $R_t > 0$ ); caso contrário, uma recompensa negativa ( $R_t < 0$ ). Recompensas negativas são dadas sempre que há um decaimento da métrica de desempenho escolhida, independente se houve uma troca de protocolos ou não. A ideia é que decaimentos indicam que o melhor protocolo de outrora possa não ser mais a melhor opção dado alguma modificação da rede.

A magnitude da recompensa é percentualmente proporcional ao aumento ou decaimento da métrica alvo numa escala de  $[-\xi, \xi]$ . Nesta dissertação, utilizamos  $\xi = 5$ . Esse valor foi obtido empiricamente, sendo grande o suficiente para rápida convergência e não grande o bastante para prender o mecanismo de seleção em mínimos locais por muito tempo. A equação 4.1 representa o sistema de recompensas utilizado. Vale ressaltar que  $g_t$  está contido no intervalo  $[-1, 1]$ . Caso o ganho (ou decaimento) percentual exceda 100%,  $g_t$  ainda será igual a 1 (ou  $-1$ ). Similar à magnitude de  $\xi$ , o valor de  $g_t$  também influencia no tempo de convergência e estabilidade da função  $Q$ . Permitir que  $g_t$  assuma valores de ordens de grandeza (e.g. de Mbps para Gbps caso  $g_t$  seja baseado na vazão) pode levar o agente a ficar preso em uma ação específica, inibindo o processo de exploração demasiadamente. Portanto, restringimos o valor absoluto de  $g_t$  a, no máximo, 1.

$$R_t = \begin{cases} 0, & \text{se } g_t > 0 \text{ e não houve mudança de protocolo} \\ g_t \times \xi, & \text{caso contrário} \end{cases} \quad (4.1)$$

$$g_t = \text{aumento ou decaimento percentual de } m \in [-1, 1] \quad (4.2)$$

As recompensas são independentes da natureza da métrica de desempenho escolhida. Por exemplo, as métricas listadas na tabela 4.1 são de diferentes naturezas e, conseqüentemente, possuem magnitudes distintas. A magnitude da recompensa  $R_t$  afeta o tempo de convergência e estabilidade da função de qualidade  $Q$ . A equação 4.1 define valores de recompensas baseados em variações percentuais da métrica de desempenho selecionada. A magnitude está contida no intervalo  $[-\xi, \xi]$  para qualquer métrica de desempenho escolhida. Assim, o sistema de recompensas é independente da natureza e magnitude da métrica de rede escolhida para a seleção do protocolo MAC.

Além disso, é importante citar que algumas métricas de desempenho devem ser minimizadas. Esse é o caso da latência, por exemplo. Nesse caso, o valor de  $g_t$  é proporcionalmente positivo à diminuição da latência e proporcionalmente negativo ao aumento. No mais, o funcionamento é igual para qualquer tipo de métrica de desempenho escolhida para definição do melhor protocolo MAC.

Por fim, vale mencionar que o cálculo da recompensa pode ser modificado para alterar o comportamento do algoritmo Q-Learning. O custo de trocas de protocolos pode diferir de acordo com o tipo de dispositivo sem fio. Por exemplo, trocas podem ter alto custo em dispositivos com recursos energéticos limitados. O custo também pode ser alto em sistemas de tempo real, onde a troca acarretaria em queda de desempenho instantâneo. Nesses casos, recompensas positivas podem ser limitadas apenas aos casos

de maior ganho de desempenho, tornando o agente menos propenso a executar trocas em situações de baixo ou nenhum ganho.

#### 4.1.2.3 Hiperparâmetros

Os valores dos hiperparâmetros de Q-Learning por estratégia de exploração estão listados na tabela 4.2. A quarta coluna corresponde aos hiperparâmetros intrínsecos de cada estratégias de exploração (capítulo 2, seção 2.3.1), sendo: a)  $\epsilon$  para  $\epsilon$ -greedy, b)  $T$  para softmax e c)  $c$  para UCB. Os valores foram obtidos empiricamente. O processo de obtenção é descrito posteriormente na seção 5.1 do próximo capítulo.

Estratégia de exploração	$\alpha$	$\gamma$	*
$\epsilon$ -greedy	0.8	0.7	0.1
softmax	0.7	0.8	0.5
UCB	0.9	0.5	2

Tabela 4.2: Hiperparâmetros por estratégia de exploração

#### 4.1.2.4 Discussão

Um assunto ainda não discutido até o momento é a inicialização da função  $Q$ . Os valores iniciais dessa função podem ser utilizados em forma de conhecimento prévio do sistema. Embora essa prerrogativa seja tentadora à primeira vista, ela não corrobora com ambientes não estacionários. Nas palavras de [Sutton and Barto, 1998], “qualquer método que foque em condições iniciais é improvável de ajudar em casos de não estacionariedade” (tradução livre). No escopo de SOMAC, é difícil especificar quaisquer valores iniciais para  $Q$  sem a rede sem fio ter entrado em operação. Dessa forma, a função  $Q$  é inicializada em zero para todos os possíveis pares de estados e ações nesta dissertação, a menos que afirmado o contrário.

Outro ponto importante é a escalabilidade da modelagem adotada para o algoritmo Q-Learning. A modelagem considera estados como protocolos MAC e ações como a troca ou a manutenção do protocolo. A adição de novos protocolos aumenta a máquina de estados do algoritmo Q-Learning (figura 4.4), sendo ambos estados e ações iguais ao número de protocolos.

O número de protocolos afeta o desempenho de SOMAC. A convergência do algoritmo Q-Learning em ambientes não estacionários é garantida pela exploração a partir de todos os pares de estados e ações  $((s, a), \forall s \in S \text{ e } a \in A)$  ao longo do tempo [Sutton and Barto, 1998]. A ordem de complexidade de Q-Learning é dada por  $Q(ev)$ , onde  $v = |S|$  e  $e = \sum_{s \in S} |A(s)|$  [Koenig and Simmons, 1992]. Como em SOMAC as

cardinalidades  $|S|$  e  $|A(s)|$  são iguais a  $n$  (onde  $n$  é o número de protocolos MAC), a ordem de complexidade do processo de exploração é assintoticamente dada por  $O(n^3)$ .

Apesar de cubicamente proporcional ao número de protocolos no pior caso, o desempenho médio pode ser satisfatório na prática. Primeiro, estratégias de exploração como softmax avaliam o potencial das ações, favorecendo as mais promissoras (capítulo 2, seção 2.3.1) e, conseqüentemente, resultando em menor tempo de convergência. Segundo, mesmo nos piores casos, o algoritmo temporariamente conduz a soluções subótimas [John, 1998]. No escopo de SOMAC, soluções subótimas representam a escolha do melhor protocolo MAC dentro de um subconjunto dos protocolos disponíveis. Conseqüentemente, a rede sem fio operará com protocolos melhores que o pior caso. Esse comportamento é mais crítico em situações de alta dinamicidade da rede sem fio, quando os melhores e piores protocolos são alternados constantemente. Dessa forma, SOMAC evita os piores protocolos mesmo que temporariamente não selecione o melhor. Em situações onde a rede é estática ou sua dinamicidade seja baixa, a transição de soluções subótimas para a solução ótima é assegurada por estratégias de exploração [Sutton and Barto, 1998]. Em todos os casos, o desempenho da rede tende ao desempenho máximo possível a termo<sup>1</sup>.

## 4.2 Implementação

A proposta SOMAC foi implementada utilizando-se rádios definidos por software. Os dispositivos utilizados para implementação e avaliação são rádios USRP da família x2xx pertencentes ao laboratório Winet<sup>2</sup> e ao *testbed* UFMG do projeto FUTEBOL [FUTEBOL, 2018].

A programação dos rádios definidos por software foi feita via a plataforma GNU Radio. O trabalho de [Cordeiro et al., 2018] serviu como base para a implementação desta proposta. As próximas seções possuem informações sobre a implementação de SOMAC no GNU Radio.

### 4.2.1 GNU Radio

GNU Radio é a plataforma utilizada nesta dissertação para programação de rádios definidos por software. O objetivo é implementar a subcamada MAC ilustrada pela figura 4.2. A subcamada consiste em dois protocolos MAC (CSMA/CA e TDMA),

---

<sup>1</sup>A convergência global nem sempre é possível, a depender do quão frequente e significativamente a rede é conturbada

<sup>2</sup><http://www.winet.dcc.ufmg.br/>

módulos de sensoriamento e um mecanismo de decisão responsável por seleccionar o protocolo MAC a ser utilizado. Adicionalmente, desenvolvemos módulos para geração e detecção das métricas de desempenho da rede (tabela 4.1).

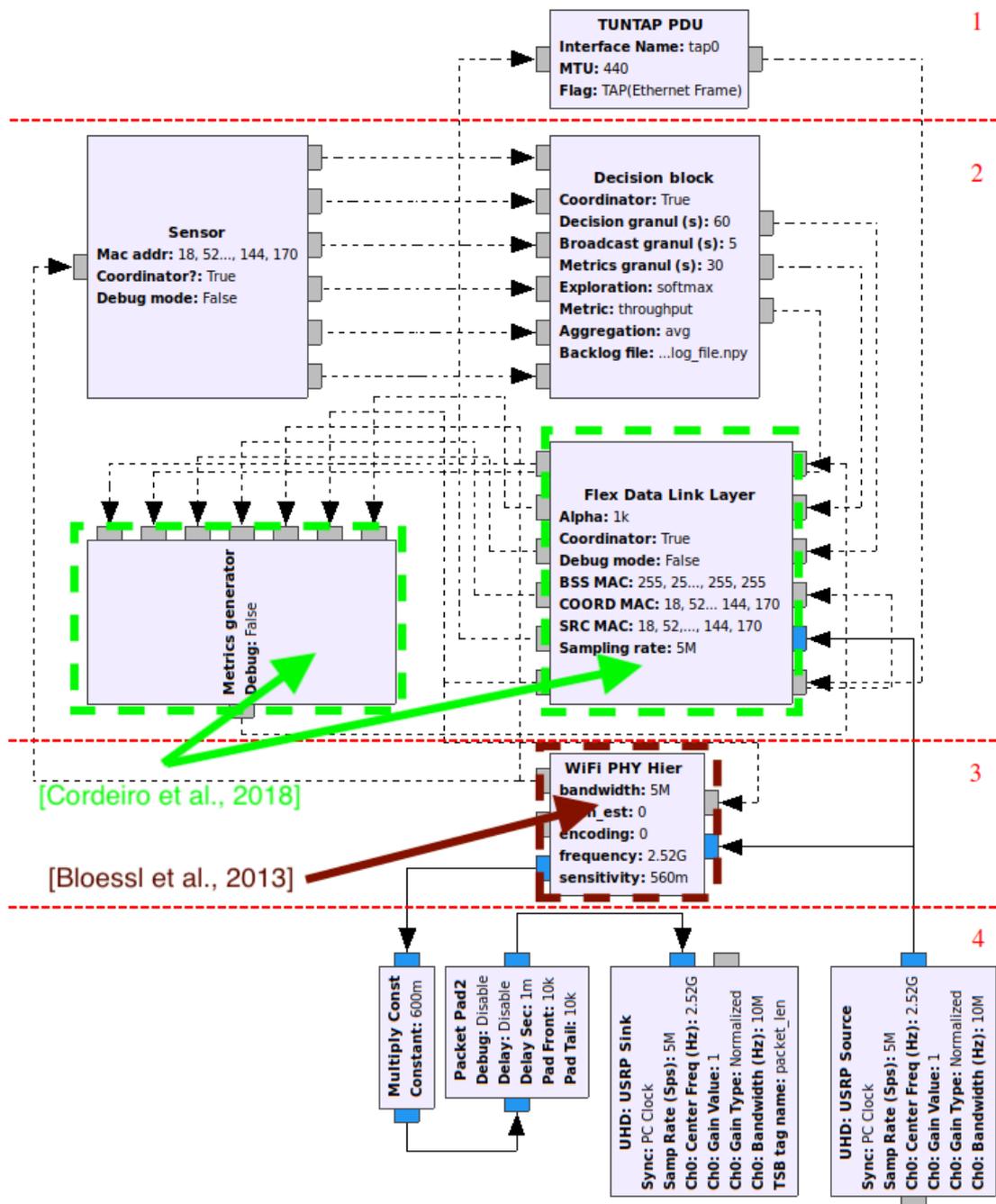


Figura 4.5: SOMAC: arquitetura de blocos na plataforma GNU Radio

A figura 4.5 ilustra os módulos de SOMAC na plataforma GNU Radio. Os blocos em verde possuem alta correlação com o trabalho de [Cordeiro et al., 2018]. Alterações

consideráveis foram feitas para otimizar e expandir o número de métricas de desempenho, e compatibilidade com os blocos "TUNTAP PDU" e "WiFi PHY Hier". Ainda assim, parte do código fonte<sup>3</sup> disponibilizado por [Cordeiro et al., 2018] foi utilizado na implementação.

Os módulos da figura 4.5 podem ser divididos em quatro grupos. O grupo 1 é formado pelo bloco "TUNTAP PDU". Esse bloco integra projetos do GNU Radio com a pilha de protocolos do sistema operacional e é um bloco nativo da plataforma GNU Radio. O "TUNTAP PDU" permite que a placa GNU Radio funcione como uma placa de rede convencional no sistema operacional, como ilustra a figura 4.6.

```
sdr@martell:~$ ifconfig tap0
tap0      Link encap:Ethernet  HWaddr 12:34:56:78:90:bd
          inet addr:192.168.123.10  Bcast:192.168.123.255  Mask:255.255.255.0
          UP BROADCAST MULTICAST  MTU:440  Metric:1
          RX packets:195 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5086962 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:19110 (19.1 KB)  TX bytes:498516138 (498.5 MB)
```

Figura 4.6: Interface de rede "TUNTAP"

Já o grupo 2 corresponde à proposta desta dissertação. Os dois blocos à direita são o mecanismo de decisão e a subcamada MAC com os protocolos CSMA/CA e TDMA, de cima para baixo respectivamente. O bloco de decisão ("Decision block") possui o algoritmo de aprendizado por reforço Q-Learning, assim como as três diferentes estratégias de exploração apresentadas anteriormente:  $\epsilon$ -greedy, softmax e UCB. O bloco é implementado em Python, incluindo os algoritmos de aprendizado por reforço. O *framework* NumPy<sup>4</sup> é utilizado na implementação do algoritmo Q-Learning assim como das estratégias de exploração. Já o bloco "Flex Data Link Layer" possui a implementação dos protocolos MAC CSMA/CA e TDMA, e do par MUX/DEMUX para a seleção do protocolo. Esse bloco é controlado pelo bloco de decisão, alterando o protocolo MAC de acordo com mensagens de controle recebidas. A implementação é feita na linguagem C++.

Ainda no grupo 2, os blocos à esquerda são responsáveis pelas métricas de rede listadas na tabela 4.1. O bloco "Metrics generator" avalia o desempenho de parâmetros ligados à camada de enlace e, periodicamente, envia um quadro com essa informação para o nó coordenador. Por sua vez, o bloco "Sensor" é responsável pela identificação dos quadros gerados e enviados pelo bloco "Metrics generator". Após extrair as métricas de rede, o bloco as repassa para o bloco de decisão caso seja o nó coordenador. É

<sup>3</sup>Disponível em <https://github.com/jeffRayneres/FS-MAC.git>

<sup>4</sup><http://www.numpy.org/>

também função do bloco "Sensor" identificar mensagens de controle, informando o protocolo MAC a ser usado. Essas mensagens são repassadas ao bloco de decisão dos nós normais para que eles operem de acordo com o protocolo definido pelo nó coordenador. Tanto "Sensor" quanto "Metrics generator" são implementados em C++.

A sobrecarga total das mensagens de controle geradas por esses blocos depende do número de nós conectados à rede. Mensagens de controle enviadas pelo nó coordenador possuem um tamanho total de 42 bytes. Isso corresponde a uma sobrecarga de 8.4 Bps considerando-se uma granularidade de 5 segundos (valor utilizado nesta dissertação). Já o tamanho dos quadros enviados pelos demais nós ao reportarem métricas de desempenho é de 150 bytes. Sendo a granularidade utilizada de 30 segundos, a sobrecarga dessas mensagens é de 5 Bps por nó escravo. Assim, a sobrecarga total da rede é de  $(8.4 + 5n)$  Bps, onde  $n$  é o número de nós escravos presentes na rede.

O grupo 3 corresponde à camada física da pilha de protocolos de rede. O bloco "WiFi PHY Hier", desenvolvido por [Bloessl et al., 2013], implementa a modulação OFDM e o enquadramento seguindo a norma IEEE 802.11. Ele foi desenvolvido para suportar comunicações nas normas IEEE 802.11 a/g/p<sup>5</sup>. Neste trabalho, adotamos, exclusivamente, o modo de operação compatível com a norma IEEE 802.11g com frequência de operação de 2 GHz, modulação BPSK (do inglês *Binary Phase Shift Keying*) com taxa de codificação  $1/2$  e taxa de amostragem de 5 Msps (*samples per second*). A taxa de amostragem reduzida possibilita o processamento do fluxo de sinais sem sobrecarregar os dispositivos disponíveis para experimentação.

Por fim, os blocos do grupo 4 representam a rede sem fio. Os blocos "UHD: USRP Sink" e "UHD: USRP Source" são *drivers* fornecidos pelo fabricante (Ettus Research<sup>6</sup>) dos dispositivos SDR USRP utilizados nesta dissertação. Ambos se referem ao mesmo rádio USRP. O bloco *sink* funciona como um canal para transmissão de dados. Os sinais são recebidos pelo bloco e, em sequência, encaminhados ao rádio USRP. Já o bloco *source* tem função oposta. Os sinais recebidos pelo rádio são encaminhados para esse bloco e, posteriormente, enviados à camada física, bloco "WiFi PHY Hier". Os demais blocos do grupo 4 fazem parte do pacote fornecido por [Bloessl et al., 2013] e são parâmetros de ajuste para utilização dos *drivers* UHD USRP.

### 4.2.2 Conformidade com a norma IEEE 802.11

A camada física da pilha de protocolos de rede utilizada na implementação de SOMAC é compatível com o padrão IEEE 802.11 [Bloessl et al., 2013]. Além disso, a implemen-

---

<sup>5</sup><https://github.com/bastibl/gr-ieee802-11>

<sup>6</sup><https://www.ettus.com/>

tação do protocolo MAC CSMA/CA também segue as especificações desse padrão, e é baseada na documentação [IEEE, 2016]<sup>7</sup>. Contudo, a implementação desta dissertação não é compatível com dispositivos comerciais sem fio que operam de acordo com o padrão IEEE 802.11.

O padrão IEEE 802.11 possui restrições de temporização difíceis de serem satisfeitas por rádios definidos por software. No escopo dos protocolos CSMA/CA e TDMA, temporizações são necessárias, por exemplo, para detecção de energia (para definir se o meio está ocupado ou não), *backoff* e sincronização de janelas de transmissão. Dois tipos de implementação em rádios definidos por software podem ser encontrados. Primeiro, implementações totalmente em software. Esse é o caso de SOMAC, onde todas as camadas da pilha de protocolos de rede são implementadas na plataforma GNU Radio (figura 4.5). Essa abordagem traz maior flexibilidade, mas tem como desvantagem atrasos introduzidos pelo processamento de sinais em uma CPU genérica com um sistema operacional não determinístico [Bloessl et al., 2014], [Schmid et al., 2007].

O segundo tipo de implementação é híbrido. Nele, a implementação é parcialmente em *hardware* e parcialmente em software. Por exemplo, [Bloessl et al., 2014] propõe um protótipo do padrão IEEE 802.11p onde funções de detecção de energia são implementadas via FPGA de rádios definidos por software, diminuindo a carga de processamento de sinais da CPU. Ainda assim, o protótipo é apenas compatível com soluções comerciais para comunicação *broadcast*.

Há soluções mais avançadas que conseguem satisfazer as restrições de temporização do 802.11. O projeto Sora possui implementação altamente otimizada e é capaz de satisfazer os requisitos de temporização do IEEE 802.11 a/b/g, interoperando com dispositivos comerciais [Tan et al., 2011]. Além disso, a comunicação entre SDR e computador é feita através de uma interface PCIe, garantindo alta performance e baixa latência. Contudo, as otimizações necessárias adicionam complexidade ao sistema, dificultando a utilização da plataforma para pesquisa e prototipagem.

Otimizações também são possíveis em rádios USRP. A família de dispositivos utilizada nesta dissertação é embarcada com FPGAs que podem ser usadas para funções como detecção de energia por exemplo, similarmente ao trabalho de [Bloessl et al., 2014]. Contudo, a FPGA já é parcialmente utilizada pelo *driver* UHD, restando poucos recursos a serem explorados [Ettus Research, 2018b]. Adicionalmente, o acesso à FPGA da família USRP x2xx (utilizadas nesta dissertação) é custoso. Ettus Research disponibiliza um *framework* de desenvolvimento chamado RFNoC [Ettus Research, nd] que facilita a utilização da FPGA, porém ele não é suportado por dispositivos USRP

---

<sup>7</sup>Os mecanismos de controle RTS e CTS são opcionais, não estando atualmente disponíveis em SOMAC.

x2xx. O *testbed* Iris do projeto [FUTEBOL, 2018] possui modelos USRP x310, compatíveis com o *framework* RFNoC. Todavia, não é possível utilizá-los para programação de FPGA, uma vez que esses dispositivos não estão ligados a um barramento serial, necessário para a programação.

Sendo assim, SOMAC não possui conformidade com dispositivos comerciais operando com o padrão IEEE 802.11 em termos de temporização até a presente data. Uma possível solução comercial pode contar com implementações em *hardware* das camadas física e de enlace, com exceção do mecanismo de decisão ("Decision block", figura 4.5). Essa abordagem seria conforme com a temporização e, portanto, igualaria o desempenho dos protocolos MAC de SOMAC ao protocolos MAC de redes sem fio comerciais.

Ainda assim, a proposta é uma prova de conceito da troca de protocolos MAC em redes sem fio. Os resultados derivados deste trabalho e discutidos no próximo capítulo mostram os benefícios dessa abordagem, sugerindo que futuras implementações alinhadas com o desempenho de padrões comerciais podem afetar positivamente no desempenho de redes sem fio.

## 4.3 Resumo

Este capítulo apresenta uma visão geral da arquitetura da solução proposta nesta dissertação. A solução é denominada SOMAC e seleciona o melhor protocolo MAC de acordo com o estado da rede sem fio. A seleção é contínua, adaptável a mudanças da rede sem fio e consegue lidar com ambientes não estacionários. Para isso, o mecanismo de decisão de SOMAC, responsável pela seleção do protocolo MAC, utiliza a técnica de aprendizado por reforço Q-Learning.

A modelagem do algoritmo Q-Learning é feita considerando-se: (1) protocolos MAC como estados e (2) manter ou trocar o protocolo como ações. A modelagem simplista facilita o processo de exploração e de aprendizado.

Por fim, SOMAC é uma implementação exclusivamente em software, sendo uma prova de conceito da viabilidade de troca de protocolos MAC baseado em aprendizado por reforço em redes sem fio.



# Capítulo 5

## Avaliação

Este capítulo apresenta os resultados para o mecanismo de troca de protocolos MAC proposto nesta dissertação. Primeiro, apresentamos uma avaliação de hiperparâmetros para cada uma das estratégias de exploração do algoritmo Q-Learning. Segundo, discutimos uma avaliação prática da melhor estratégia de exploração, levando-se em consideração diferentes cenários de avaliação, acurácia do mecanismo de seleção e ganho real em redes sem fio.

### 5.1 Análise e ajuste de hiperparâmetros

A técnica de aprendizado por reforço Q-Learning é regida pela equação 5.1. Nela, as variáveis  $\alpha$  e  $\gamma$  são hiperparâmetros que ditam a taxa de aprendizado e a taxa de desconto de futuras ações, respectivamente. Além disso, as diferentes estratégias de exploração  $\epsilon$ -greedy, softmax e UCB também possuem hiperparâmetros a serem ajustados, sendo  $\epsilon$ ,  $T$  e  $c$  respectivamente.

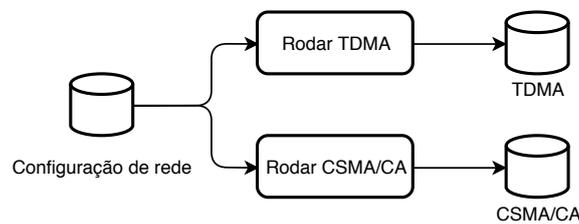
$$Q(S_t, A_t) \leftarrow (1 - \alpha)Q(S_t, A_t) + \alpha[R_{t+1} + \gamma \max_a Q(S_{t+1}, a)] \quad (5.1)$$

Esta seção trata de uma análise empírica desses hiperparâmetros. A próxima seção descreve a metodologia utilizada para a obtenção dos valores dos hiperparâmetros, seguida de uma análise e discussão dos resultados obtidos.

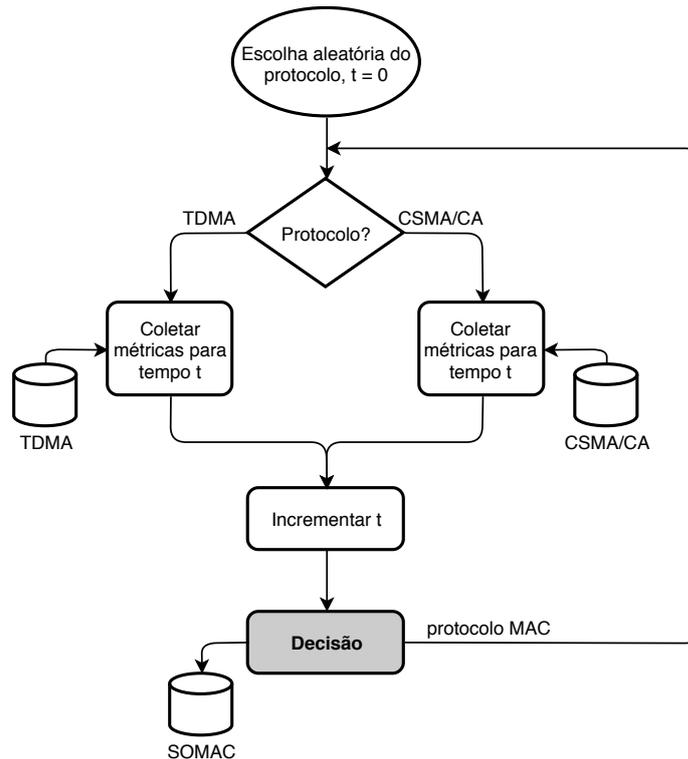
#### 5.1.1 Metodologia

A análise dos hiperparâmetros é feita através de análises numéricas da rede sem fio. Os protocolos MAC CSMA/CA e TDMA são executados em redes sem fio reais e, para cada configuração de rede, as métricas de desempenho de ambos os protocolos

são obtidas. A figura 5.1a ilustra esse processo. Em seguida, os dados obtidos são utilizados para alimentar o módulo de decisão de SOMAC conforme ilustrado pelo fluxograma da figura 5.1b. Dessa forma, o módulo de decisão recebe como entrada métricas do protocolo selecionado naquele momento, similar ao funcionamento real de SOMAC. O protocolo inicial é aleatoriamente escolhido. O algoritmo 4 retrata esse processo com maior detalhamento. No caso de trocas de protocolos, a métrica recebida pelo módulo de decisão é uma média aritmética entre a métrica do protocolo anterior e do protocolo escolhido. Isso representa eventuais quedas de desempenho durante transições, penalizando múltiplas trocas seguidas.



(a) Coleta de dados



(b) Análise do módulo de decisão

Figura 5.1: Fluxograma da análise numérica de SOMAC

As métricas de desempenho dos protocolos CSMA/CA e TDMA são obtidas experimentalmente. Os valores correspondem a diferentes configurações de 10 rádios

**Algoritmo 4** Análise do funcionamento de SOMAC

---

**Entrada**  $arr_{csmac}[0...T]$ ,  $arr_{tdma}[0...T]$   
 Inicializar  $Q \leftarrow \emptyset$   
 Inicializar  $protocolo \leftarrow random(csmac, tdma)$   
 $arr_{samac}[0] \leftarrow arr_{csmac}[0]$  se  $protocolo$  é  $csmac$ , senão  $arr_{tdma}[0]$   
**para**  $t \leftarrow 0$  **até**  $T$  **faça**  
    $p \leftarrow SOMAC(arr_{samac}[t])$   $\triangleright$  Seleção do protocolo com decisão de SOMAC  
   **se**  $p$  não é  $protocolo$  **então**  $\triangleright$  Troca de protocolos  
      $arr_{samac}[t+1] \leftarrow avg(arr_p[t], arr_{protocolo}[t+1])$   
   **senão**  
      $arr_{samac}[t+1] \leftarrow arr_{csmac}[t+1]$  se  $protocolo$  é  $csmac$ , senão  $arr_{tdma}[t+1]$   
   **fim se**  
    $protocolo \leftarrow p$   
   Computar recompensa e atualizar matriz  $Q$   
**fim para**

---

USRP, sendo 8 do *testbed* UFMG [FUTEBOL, 2018] e 2 rádios extras do laboratório Winet<sup>1</sup>. Os nós do *testbed* UFMG consistem em rádios USRP modelos B200 e B210 conectados à computadores Intel Core i7-6700T com 16GB de memória RAM e sistema operacional Ubuntu 16.04 LTS, enquanto os nós extras consistem em computadores Intel Core i5-3317U com 8 GB de memória RAM e sistema operacional Ubuntu 16.04 LTS, conectados a rádios USRP B210. A figura 5.2 ilustra a disposição dos rádios USRP com os respectivos números de identificação, sendo "C" o nó coordenador.

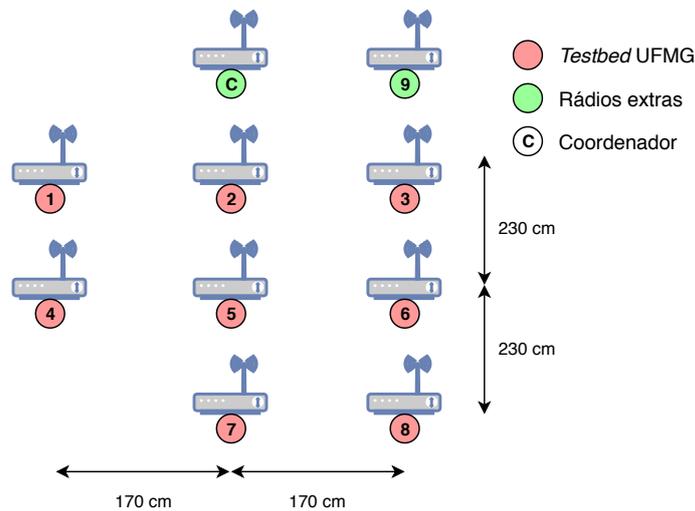
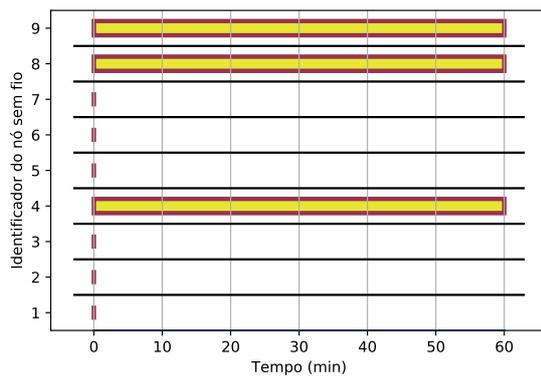


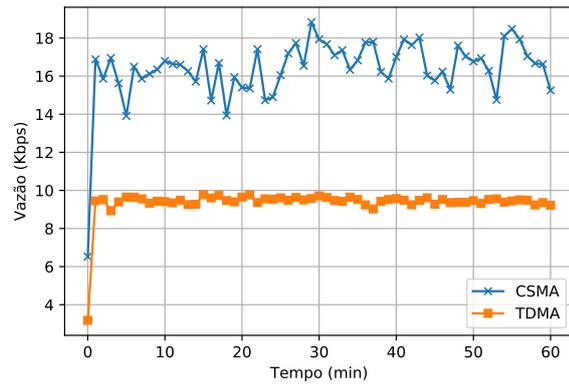
Figura 5.2: Disposição dos rádios USRP do *testbed* UFMG

Dados de três configurações de redes diferentes são utilizadas. Cada configuração consiste em um arranjo de tempos de transmissão e silêncio dos nós ilustrados na figura

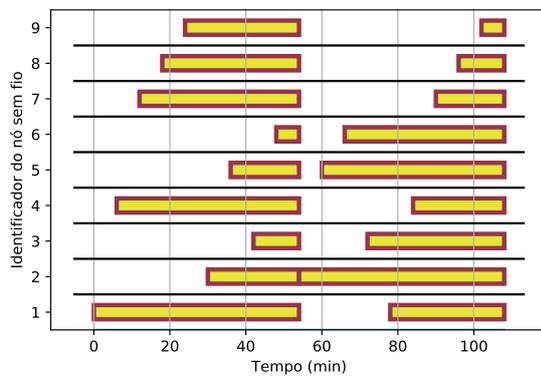
<sup>1</sup>Laboratório responsável pelo *testbed* UFMG, <http://www.winet.dcc.ufmg.br/>



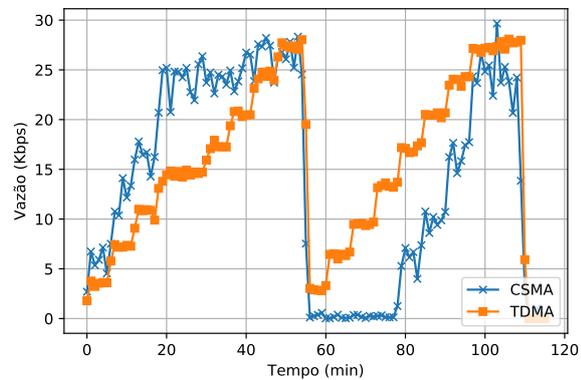
(a) LT: config. estática



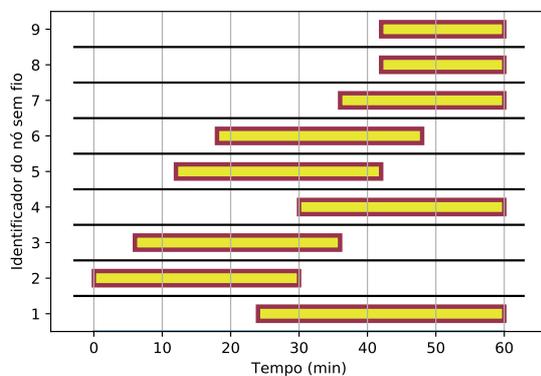
(b) D: config. estática



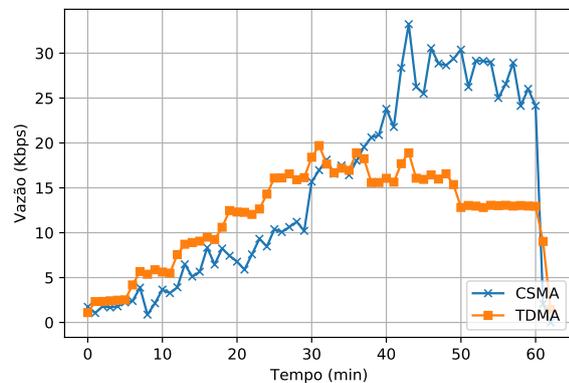
(c) LT: config. com troca bem definida



(d) D: config. com troca bem definida



(e) LT: config. com troca gradual



(f) D: config. com troca gradual

Figura 5.3: Configurações de rede

5.2, com exceção do nó coordenador que permanece ativo durante todo o tempo. As figuras 5.3a, 5.3c e 5.3e mostram as linhas do tempo (LT) das configurações de redes avaliadas, onde as barras horizontais representam os momentos em que cada nó está transmitindo quadros ao longo do tempo. O tráfego consiste em pacotes ICMP de

64 bytes gerados pela ferramenta *ping*<sup>2</sup> a uma taxa de 100 pacotes por segundo, de modo a saturar a capacidade de transmissão de cada nó quando estes estão ativos. Similarmente, as figuras 5.3b, 5.3d e 5.3f ilustram o desempenho (D) dos protocolos CSMA/CA e TDMA das respectivas configurações em termos da vazão. Essas configurações foram manualmente montadas com o intuito de avaliar o desempenho do módulo de decisão em situações específicas de redes, variando de cenários estáticos (figs. 5.3a, 5.3b), cenários com ponto de troca de protocolos bem definido (figs. 5.3c, 5.3d, com troca entre os minutos 50 e 60) a cenários com variação gradual (figs. 5.3e, 5.3f).

O objetivo é comparar diferentes valores de hiperparâmetros com o intuito de obter os que resultam em melhor desempenho. A avaliação de desempenho é feita através do *regret* médio, equação 5.2. O *regret* corresponde à diferença entre o desempenho ótimo e o desempenho obtido [Garivier and Moulines, 2011]. O desempenho ótimo é dado pela sequência de ações que resultam no melhor protocolo MAC seguindo a política de escolha de ações  $\pi^*$ . No contexto de SOMAC, a política  $\pi^*$  é obtida comparando-se os desempenhos dos protocolos CSMA/CA e TDMA a cada instante de tempo  $t$ . Por exemplo, CSMA/CA é dito o melhor protocolo no instante  $t$  se o desempenho de CSMA/CA for superior ao desempenho de TDMA naquele instante. Já  $r(s, a|\pi)$  representa o desempenho obtido seguindo a política de escolha de ações definidas por SOMAC,  $\pi$ . Sendo assim, quanto menor o *regret*, mais próximo do desempenho ótimo.

$$\eta = \frac{1}{T} \sum_t^T r_t(s, a|\pi^*) - r_t(s, a|\pi) \quad (5.2)$$

### 5.1.2 Resultados

A figura 5.4 ilustra o *regret* percentual médio ( $\eta$  (%)) de SOMAC com estratégia de exploração  $\epsilon$ -greedy, normalizado em função da maior vazão. Os valores de  $\alpha$  e  $\gamma$  correspondem, respectivamente, às taxas de aprendizado e desconto da equação 5.1. O resultado de cada conjunto de hiperparâmetros é uma média de 30 análises por configuração de rede da figura 5.3. A estratégia de exploração  $\epsilon$ -greedy possui melhor desempenho para valores pequenos de  $\epsilon$ . De fato, os resultados das análises mostram um melhor desempenho quando  $\epsilon = 0.1$ . Além disso, o menor valor percentual de  $\eta$  foi obtido quando  $\alpha = 0.8$  e  $\gamma = 0.7$ , figura 5.4a.

Já a figura 5.5 ilustra o *regret* percentual médio em função dos hiperparâmetros de SOMAC com estratégia de exploração softmax. Os melhores desempenhos são

<sup>2</sup><https://www.unix.com/man-page/all/1M/ping/>

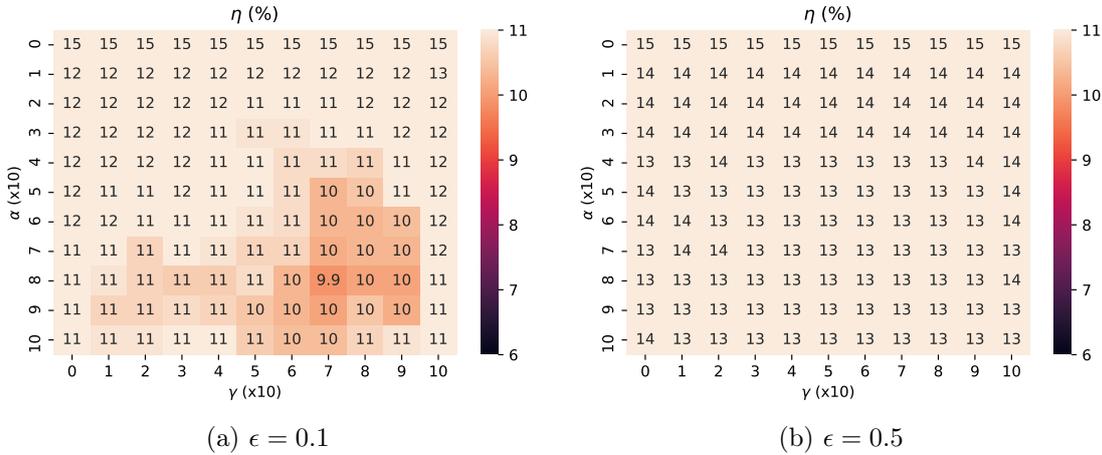


Figura 5.4:  $\epsilon$ -greedy, mapas de calor de  $\eta$  em função de  $\alpha$ ,  $\gamma$  e  $\epsilon$

obtidos quando  $T = 0.5$ . Os valores selecionados de  $\alpha$  e  $\gamma$  são, respectivamente, 0.7 e 0.8. Esses valores não resultam necessariamente no menor *regret*. Todavia,  $\alpha = 0.7$  e  $\gamma = 0.8$  estão cercados dos menores valores de  $\eta$ . Assim, pequenas variações dos hiperparâmetros não afetam o desempenho consideravelmente, indicando uma região de estabilidade.

A figura 5.6 mostra os resultados para a estratégia de exploração UCB. Nesse caso, os valores selecionados foram  $\alpha = 0.9$ ,  $\gamma = 0.5$  e  $c = 2$ . Similar ao softmax, a escolha dos hiperparâmetros de UCB também levou em consideração a região como forma de avaliar a estabilidade da escolha.

Por fim, a figura 5.7 ilustra o *regret* médio normalizado total das diferentes estratégias de exploração com os respectivos hiperparâmetros ajustados. O intervalo de confiança utilizado é de 90%. Softmax possui o menor *regret* médio, sendo ligeiramente melhor que a estratégia UCB. Por outro lado, a estratégia  $\epsilon$ -greedy é a de pior desempenho.

### 5.1.3 Discussão

As diferentes estratégias de exploração resultam em desempenho similar para os diferentes cenários analisados nesta sessão. Ainda assim, a figura 5.7 resalta um ligeiro melhor desempenho da estratégia softmax. Esse resultado corrobora com o trabalho de [Tijmsma et al., 2016], onde os autores comparam o desempenho de diferentes estratégias de exploração. Nele, softmax apresenta os melhores resultados, seguido de UCB e  $\epsilon$ -greedy.

UCB e  $\epsilon$ -greedy possuem algumas desvantagens. Primeiro, o tempo de execução

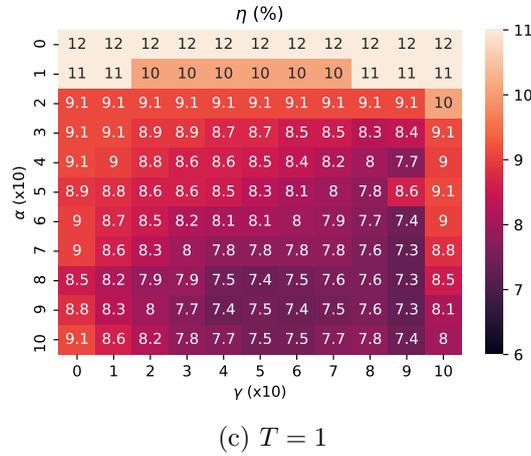
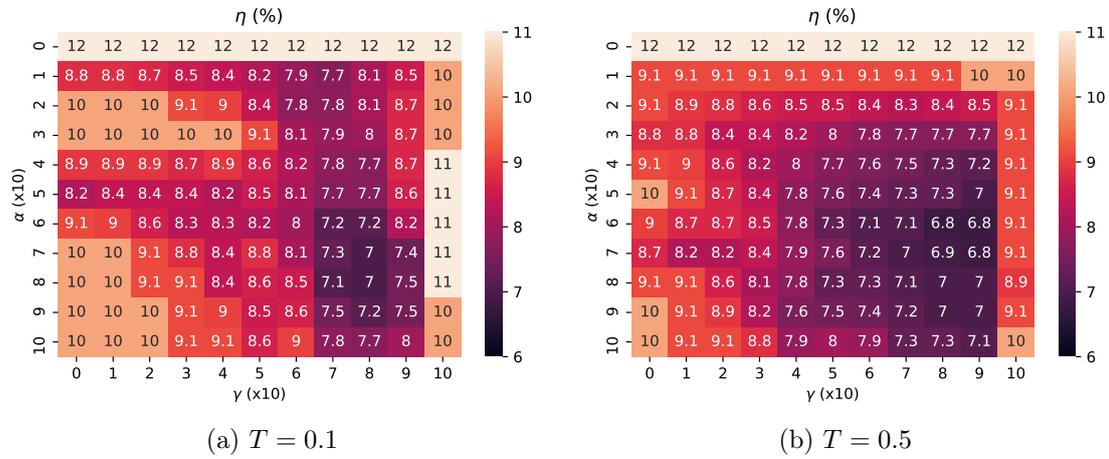


Figura 5.5: Softmax, mapas de calor de  $\eta$  em função de  $\alpha$ ,  $\gamma$  e  $T$

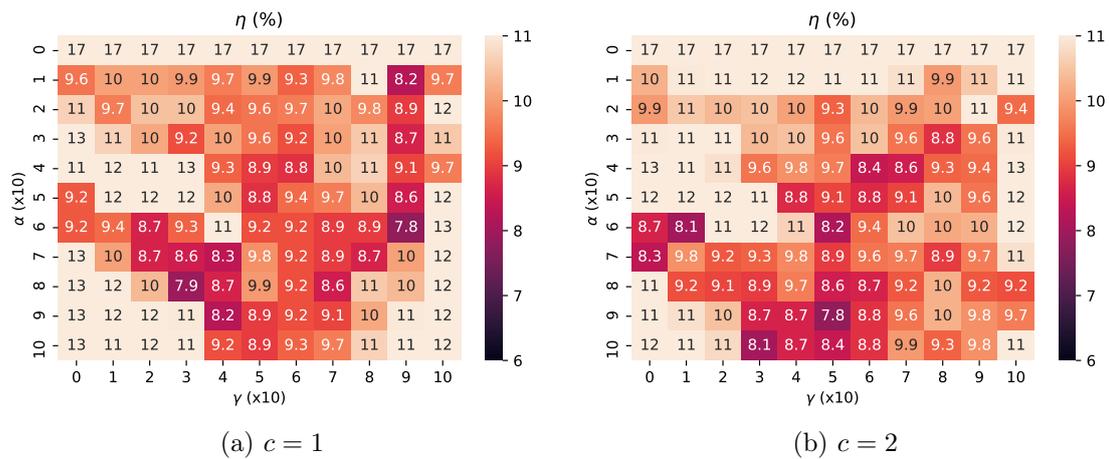


Figura 5.6: UCB, mapas de calor de  $\eta$  em função de  $\alpha$ ,  $\gamma$  e  $c$

total não é conhecido. O fator de *bonus* de UCB se comporta de forma diferente ao

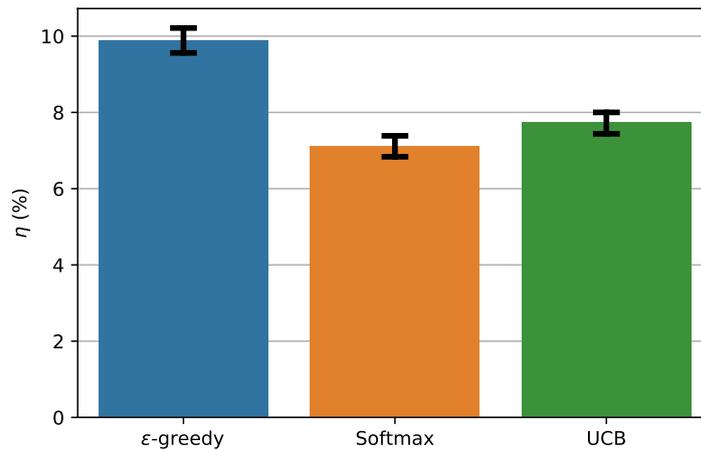


Figura 5.7: Resultado geral de desempenho das estratégias de exploração com os hiperparâmetros ajustados

longo do tempo, sendo necessário reiniciá-lo para assegurar o balanço do dilema entre exploração e exploração [Cesa-Bianchi and Lugosi, 2006]. O período de reinicialização é um outro hiperparâmetro necessário para execuções longas. Segundo,  $\epsilon$ -greedy explora ações de forma uniformemente aleatória [Sutton and Barto, 1998]. Durante a exploração, não há distinção entre ações de maior ou menor potencial. Essa abordagem pode levar a escolhas sucessivas de más ações, justificando o pior desempenho de  $\epsilon$ -greedy na figura 5.7.

Por outro lado, softmax não requer períodos de reinicialização e toma decisões exploratórias baseadas no potencial das possíveis ações, seguindo a distribuição probabilística de Boltzman (capítulo 2, seção 2.3.1). Esses fatores, o trabalho de [Tijssma et al., 2016] e os resultados ilustrados pela figura 5.7 indicam softmax como a melhor estratégia de exploração. Assim, esta dissertação avalia o desempenho de SOMAC com softmax experimentalmente. O processo de avaliação, assim como os resultados, são apresentados nas próximas seções deste capítulo.

## 5.2 Avaliação experimental

Esta seção apresenta os resultados da avaliação experimental de SOMAC com estratégia de exploração softmax. Os valores dos hiperparâmetros são os definidos na seção anterior, sendo:  $\alpha = 0.7$ ,  $\gamma = 0.8$  e  $T = 0.5$ .

### 5.2.1 Metodologia

O objetivo da avaliação experimental é avaliar o quão SOMAC é capaz de selecionar o melhor protocolo MAC ao longo do tempo. Para isso, os protocolos disponíveis rodam em configurações de rede previamente geradas de forma similar ao ilustrado pela figura 5.8. Novamente, configurações de rede correspondem à arranjos de tempos de transmissão e silêncio dos nós sem fio, cujo tráfego é gerado pela aplicação *ping* – pacotes ICMP de 64 bytes com granularidade de 10 ms. O resultado de cada execução é armazenado para análise posterior. As avaliações de desempenho de SOMAC ocorrem tanto em comparação com os protocolos CSMA/CA e TDMA quanto em relação aos trabalhos relacionados. Cada avaliação consiste em 10 execuções para cada um dos protocolos, sendo os resultados apresentados com intervalos de confiança de 90%, a menos que dito o contrário. A escolha do intervalo leva em consideração o aspecto experimental da avaliação, cujas amostras estão sujeitas a intempéries esporádicas.

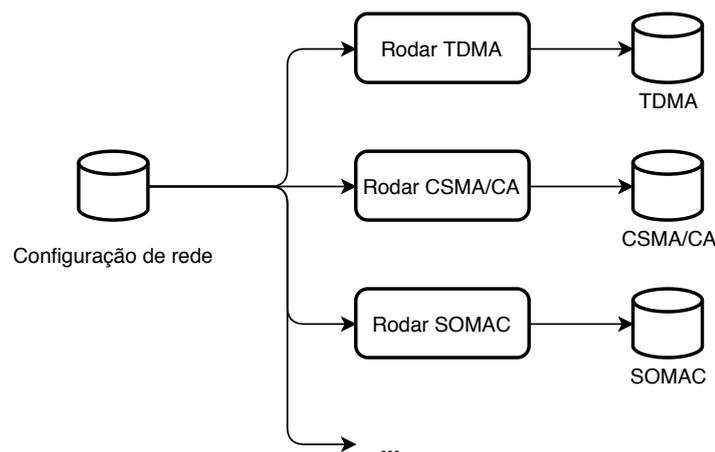


Figura 5.8: Visão geral da avaliação experimental

Novamente, o ambiente de avaliação é o *testbed* UFMG. A disposição dos rádios SDR é a mesma da figura 5.2. Os equipamentos também são os mesmos, 8 rádios USRP modelos B200 e B210 conectados a computadores Intel Core i7-6700T com 16GB de memória RAM e sistema operacional Ubuntu 16.04 LTS mais 2 rádios USRP B210 extras, cada qual conectada a um computador Intel Core i5-3317U com 8GB de memória RAM e sistema operacional Ubuntu 16.04 LTS.

A avaliação de desempenho é baseada em três métricas distintas. A primeira é dita *otimalidade* e representa o percentual de tempo que SOMAC (ou demais) operou de acordo com o protocolo MAC ótimo. Por exemplo, se SOMAC opera com otimalidade de 90%, houve interseção entre o desempenho de SOMAC e dos respectivos melhores protocolos MAC durante 90% do tempo. Nesse caso, quanto maior o valor de

otimalidade, melhor. A segunda métrica é o *regret* introduzido na seção 5.1, equação 5.2. Quanto menor o *regret*, mais próximo do desempenho ótimo. Por fim, o desempenho da rede sem fio quando empregando o SOMAC, onde avaliamos vazão, latência ou quaisquer outras métricas que possam ser utilizadas para o cálculo da recompensa (definida no capítulo 4, equação 4.1).

Como mencionado anteriormente, o desempenho do SOMAC é comparado ao desempenho do estado da arte. Selecionamos dois trabalhos relacionados de troca de protocolos para comparação de desempenho, **AMAC** [Huang et al., 2009] e **FS-MAC** [Cordeiro et al., 2018]. O trabalho de [Ye et al., 2016] não é avaliado devido às expressões matemáticas utilizadas para definição dos chamados pontos de troca. Essas expressões são específicas de dois protocolos MAC não disponíveis em SOMAC. Além disso, [Ye et al., 2016] não considera características do meio de propagação, sendo improvável de funcionar em redes sem fio reais. Já [Qiao et al., 2016] não é avaliado contra SOMAC devido à dificuldade de obtenção do conjunto de treinamento, cuja rotulação dos dados exige a repetição sistemática de configurações de rede. Adicionalmente, tanto [Qiao et al., 2016] quanto [Gonçalves et al., 2018] dependem exclusivamente do conhecimento obtido do conjunto de treinamento, não lidando com ambientes não estacionários como redes sem fio.

O protocolo MAC inicial é aleatoriamente escolhido para cada experimento quando rodando SOMAC, AMAC ou FS-MAC. O objetivo é não beneficiar um ou outro protocolo em configurações de rede específicas, o que enviesaria os resultados. Por fim, a função  $Q$  é inicializada em zero para todos os pares de ação-estado. Assim como discutido na seção 4.1.2.4 do capítulo 4, soluções com foco em valores iniciais tendem a não ter um bom desempenho em ambientes não estacionários. Desse modo, optamos por uma inicialização neutra, não favorecendo um protocolo específico logo após a inicialização do experimento.

## 5.2.2 Resultados

As próximas seções apresentam os resultados da avaliação experimental. Os resultados são agrupados em diferentes categorias de forma a facilitar o entendimento. As primeiras seções focam em mostrar o quão SOMAC é capaz de escolher o melhor protocolo, comparando-o sistematicamente ao desempenho dos protocolos CSMA/CA e TDMA. Por fim, a última seção foca no desempenho de SOMAC em relação ao estado da arte.

### 5.2.2.1 Resultados – Topologias predefinidas

O primeiro conjunto de experimentos consiste em configurações de rede manualmente definidas. Essas configurações foram empiricamente obtidas a fim de representar situações que variam desde redes estáticas a redes dinâmicas, com múltiplos pontos de troca (PT) de protocolos. Um ponto de troca representa o momento em que um protocolo MAC passa a ter melhor desempenho em relação ao outro, fazendo-se desejável a troca de protocolos. A métrica de desempenho de rede utilizada é a vazão média.

As linhas do tempo da figura 5.9 representam as mesmas configurações utilizadas para o ajuste de hiperparâmetros. Como descrito na seção 5.1, o ajuste é feito via análise numérica dos resultados de CSMA/CA e TDMA para as respectivas configurações de rede. O objetivo aqui é avaliar o desempenho de SOMAC para essas mesmas configurações de rede, mas com SOMAC operando diretamente na rede sem fio. O desempenho de SOMAC para essas configurações também é ilustrado pela figura 5.9. Em 5.9b, SOMAC tende a acompanhar o protocolo CSMA/CA com exceção dos primeiros minutos. O desempenho mediano de SOMAC até o minuto 7, aproximadamente, está relacionado à escolha aleatória do protocolo MAC inicial. Ainda assim, SOMAC atinge 90% de otimalidade nesse caso. Já o gráfico 5.9d ilustra o desempenho de SOMAC quando há um ponto de troca bem definido entre os minutos 50 e 60, circulado em vermelho. SOMAC tende a acompanhar o melhor protocolo, atingindo 97% de otimalidade. Por último, o desempenho de SOMAC com um ponto de troca gradual, também circulado em vermelho na figura 5.9f. Entre os minutos 0 e 40, SOMAC tende a acompanhar o protocolo TDMA, melhor protocolo MAC durante esse intervalo. Após o minuto 40, o desempenho de CSMA/CA se sobrepõe ao TDMA. Ainda que a convergência seja gradual até o minuto 50, SOMAC possui melhor desempenho que o protocolo TDMA, resultando em 81% de otimalidade.

Ainda que esses resultados demonstrem um bom desempenho de SOMAC, eles podem ser enviesados uma vez que foram utilizados durante a fase de ajuste de hiperparâmetros. Por outro lado, a linha do tempo da figura 5.10a mostra uma configuração de rede distinta das anteriores e não utilizada durante o processo de obtenção dos hiperparâmetros. O resultado dessa configuração é ilustrado pela figura 5.10b. Há dois pontos de troca nesse caso (destacados no gráfico), correspondendo aos minutos 20 e 60, aproximadamente. O desempenho de SOMAC tende a acompanhar o melhor protocolo ao longo do tempo embora a convergência seja gradual após o minuto 60. Isso pode ser explicado pelas grandes punições recebidas entre os minutos 25 e 60 ao selecionar-se o protocolo CSMA/CA. Nesse intervalo, TDMA é significativamente melhor que CSMA/CA, fazendo com que SOMAC receba punições máximas no caso de

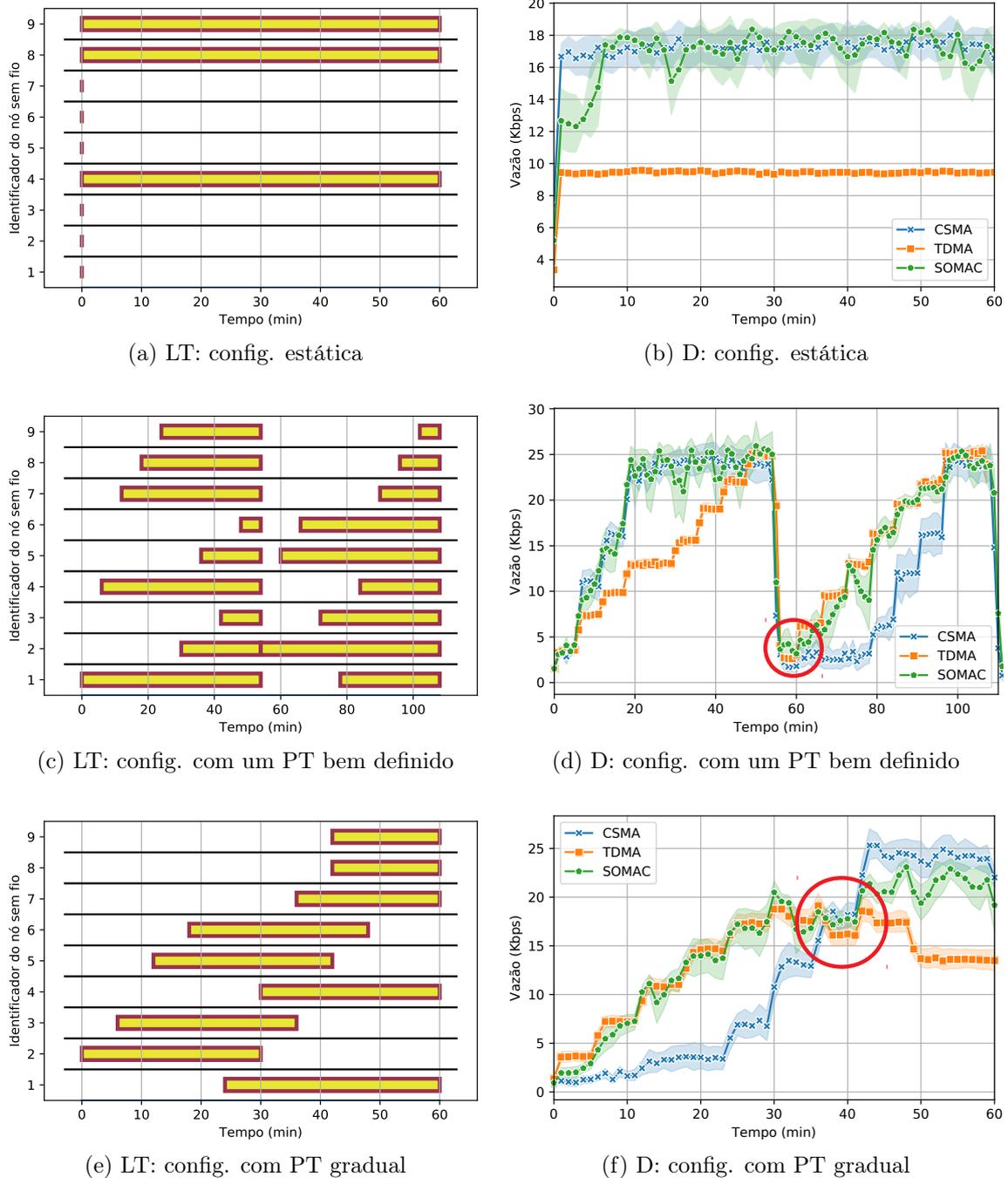


Figura 5.9: Topologias predefinidas

uma decisão errada. Consequentemente, a seleção do protocolo CSMA/CA é improvável logo após o minuto 60. Todavia, o desempenho de SOMAC tende ao desempenho de CSMA/CA com o passar do tempo, atingindo 80% de otimalidade.

Por fim, a figura 5.11 resume o desempenho do SOMAC para as quatro configurações de rede discutidas acima. Tanto a vazão quanto o *regret* são normalizados em função da maior vazão obtida. O desempenho final de SOMAC é superior a 80%

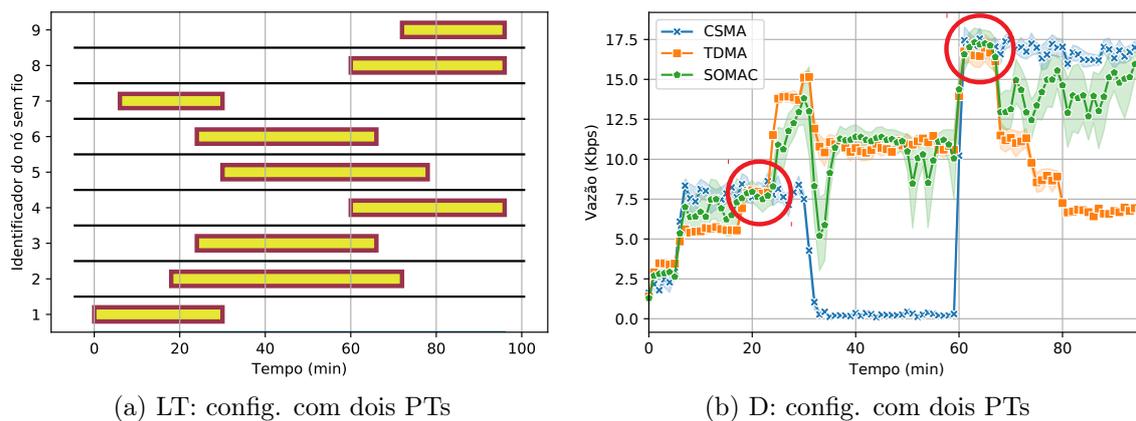


Figura 5.10: Topologia predefinida com dois pontos de troca

tanto em termos de otimalidade quanto de vazão normalizada como mostra o gráfico, superando ambos CSMA/CA e TDMA. Similarmente, SOMAC possui o menor *regret* dos três, reforçando o melhor desempenho da solução.

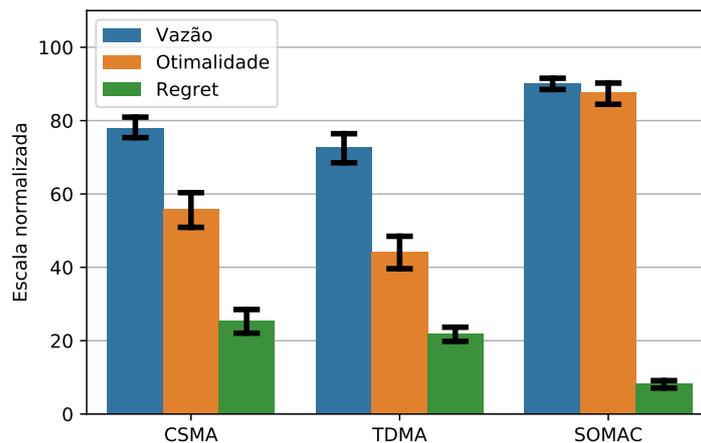


Figura 5.11: Desempenho geral para topologias predefinidas

### 5.2.2.2 Resultados – Topologia aleatória

A seção anterior apresenta resultados para configurações de rede variando de redes estáticas a redes dinâmicas. Essas configurações foram manualmente montadas para representar diferentes cenários de redes. Apesar disso, uma abordagem mais realista é a análise de desempenho de SOMAC em uma configuração aleatória. Esse cenário é ilustrado pela linha do tempo da figura 5.12a enquanto o resultado final é mostrado pela figura 5.12b. Nesse caso, CSMA/CA tende a ser o melhor protocolo MAC, excluindo-se o intervalo entre os minutos 50 e 60 (setas vermelhas do gráfico 5.12b). O desempenho de SOMAC é análogo ao desempenho de CSMA/CA, exceto naquele intervalo, quando

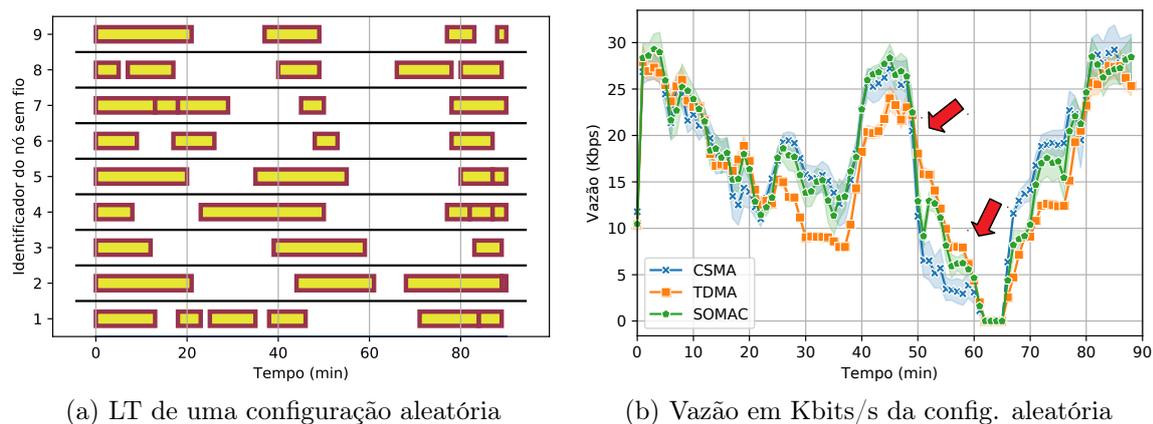


Figura 5.12: Topologia aleatória

SOMAC se assemelha ao protocolo TDMA. Como consequência, SOMAC atinge 84% de otimalidade, valor similar aos valores obtidos na seção anterior.

### 5.2.2.3 Resultados – Latência

Os resultados acima focam na seleção do protocolo MAC baseado na vazão média de dados mas essa não é a única métrica de rede possível em SOMAC. Outras métricas tais como latência, taxa de erros, jitter ou até mesmo combinações de métricas distintas podem ser utilizadas. De fato, o único requisito é que o desempenho da rede seja, de alguma forma, representado por um valor escalar, indicando melhorias ou pioras no desempenho da rede para cálculo da recompensa.

O objetivo dessa seção é verificar a flexibilidade de SOMAC em relação a uma métrica de desempenho de rede diferente da vazão, sendo a latência média a métrica escolhida. A figura 5.13 ilustra os resultados para uma configuração manualmente definida, onde há um ponto de troca de protocolos por volta do minuto 35 (seta vermelha do gráfico 5.13b). É importante ressaltar a escala logarítmica do eixo vertical do gráfico 5.13b. Ambos CSMA/CA e TDMA possuem desempenho similar entre os minutos 0 e 35, explicando a instabilidade de SOMAC durante esse intervalo, uma vez que não há grandes incentivos para a seleção de um protocolo em relação ao outro. Ainda assim, SOMAC tende a acompanhar o protocolo CSMA/CA significativamente durante esse intervalo. Por outro lado, SOMAC possui comportamento estável após o minuto 45, operando conforme o protocolo TDMA. A otimalidade final é de 84% nesse caso.

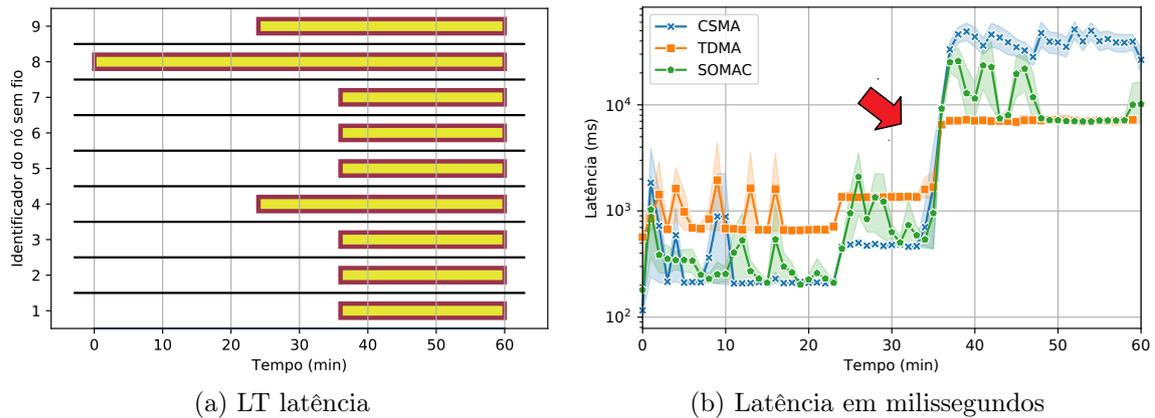


Figura 5.13: Recompensa baseada na latência média

#### 5.2.2.4 Resultados – Comparação com o estado da arte

As seções anteriores avaliam a capacidade de SOMAC de selecionar o melhor protocolo ao longo do tempo. A solução foi testada em diferentes configurações de rede, variando de uma rede estática até uma configuração aleatória. Adicionalmente, SOMAC também foi avaliado utilizando-se uma métrica de desempenho diferente da vazão de dados. Em todos esses casos, SOMAC tende a ter desempenho análogo ao melhor protocolo MAC ao longo do tempo. Nesta seção, por outro lado, SOMAC é comparado ao estado da arte. Os mesmos padrões de troca estabelecidos por [Cordeiro et al., 2018] e [Huang et al., 2009] foram utilizados durante a avaliação, conservando-se as regras de inferência de FS-MAC e o limiar de troca de 20% de AMAC.

A figura 5.14 ilustra o desempenho de SOMAC, FS-MAC e AMAC para as três primeiras configurações de rede predefinidas abordadas na seção 5.2.2.1, as mesmas utilizadas para ajuste de hiperparâmetros. Para a configuração estática (figura 5.14a), o desempenho de SOMAC é visivelmente superior ao desempenho de seus competidores. Nesse caso, tanto FS-MAC quanto AMAC tendem a seguir regras fixas de troca, ficando presos a um protocolo quando não há variações na rede, mesmo que esse protocolo não seja o de melhor desempenho. Ainda que menos discrepante, é possível notar um melhor desempenho de SOMAC nos demais gráficos de desempenho, figuras 5.14b-c.

Apesar de mostrar um bom resultado, a figura 5.14 apresenta resultados enviesados uma vez que essas configurações foram utilizadas durante o ajuste de hiperparâmetros do algoritmo Q-Learning, seção 5.1. Por outro lado, a figura 5.15 compara SOMAC aos seus concorrentes nas demais configurações apresentadas na seção 5.2.2.2, não utilizadas durante a fase de ajuste de hiperparâmetros. Em ambos os casos (i.e. gráficos 5.14a-b), o desempenho de SOMAC é igual ou melhor ao desempenho de seus

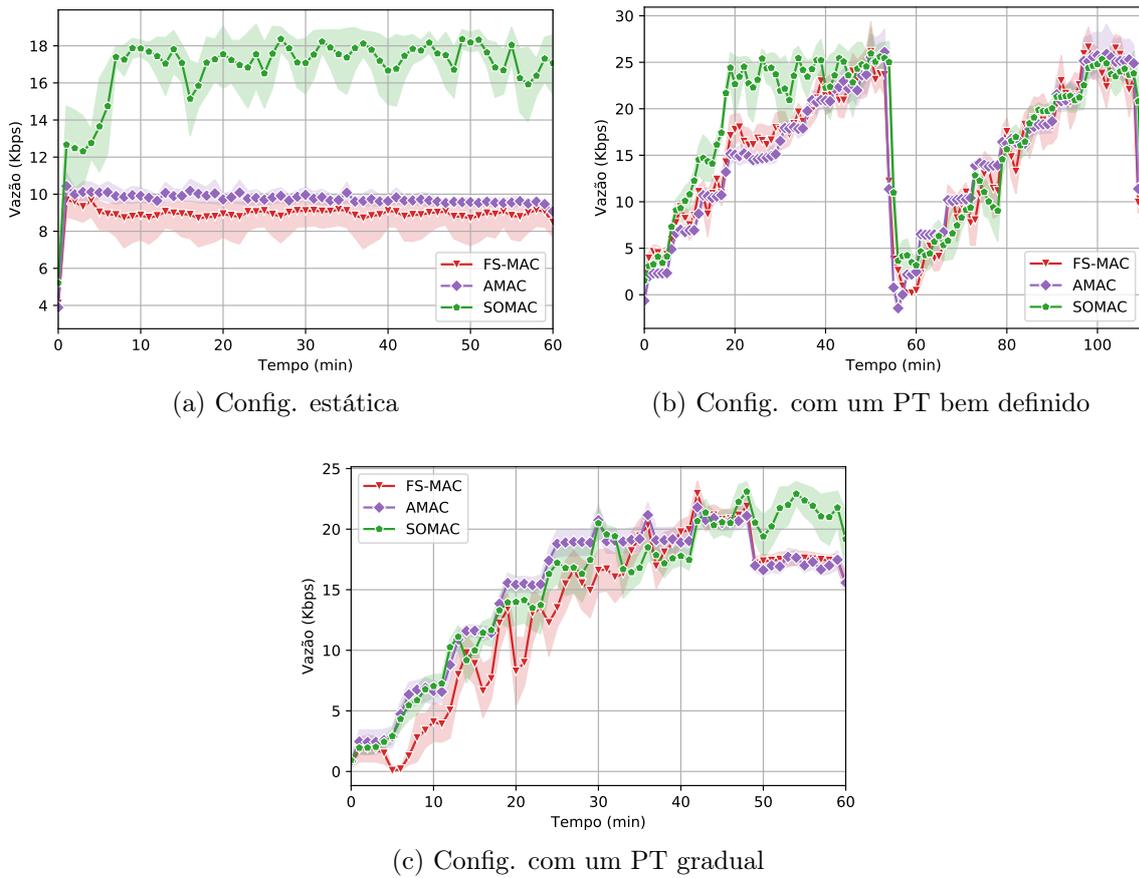


Figura 5.14: Comparação com o estado da arte em topologias predefinidas

concorrentes. Na configuração aleatória (figura 5.14b), o desempenho dos três mecanismos de troca é similar, com exceção dos intervalos [25, 35] e [40, 50] (setas vermelha e amarela, respectivamente), onde o desempenho de SOMAC se sobrepõe aos demais.

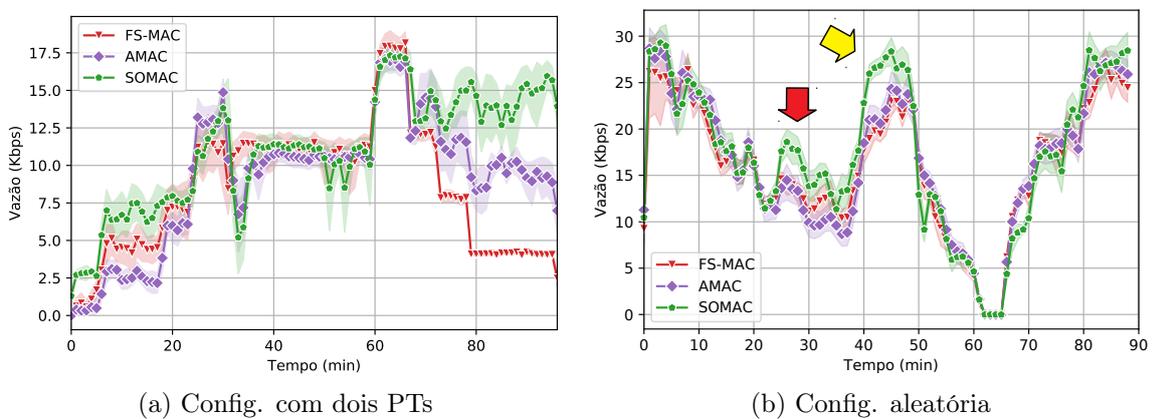


Figura 5.15: Comparação com o estado da arte em topologias com 2 PTs e aleatória

A figura 5.16 resume os resultados das figuras 5.15a-b. SOMAC é a solução com

maior número de trocas como ilustrado pela figura 5.16a. Esse comportamento parece beneficiar SOMAC em relação aos seus concorrentes uma vez que a figura 5.16b indica um melhor desempenho geral de SOMAC, resultando na maior otimalidade e no menor *regret* (normalizado em função da maior vazão). Em termos de otimalidade, SOMAC é aproximadamente 25% melhor que ambos AMAC e FS-MAC. Já em termos de *regret*, as diferenças são de, aproximadamente, 11% e 19%, respectivamente.

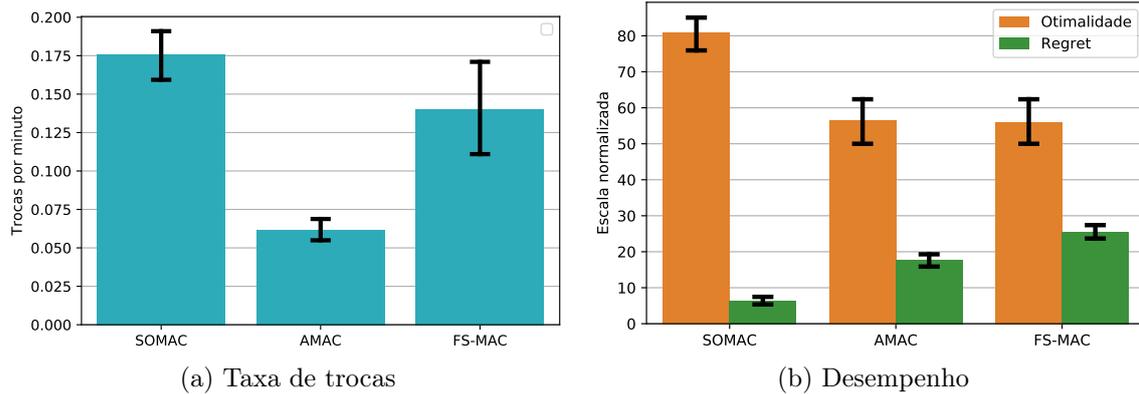


Figura 5.16: Comparação com o estado da arte em configurações com 2 PTs e aleatória

Topologias predefinidas podem beneficiar um algoritmo em relação aos demais. Mesmo uma configuração aleatória como a representada pela figura 5.15b pode possuir viés e beneficiar SOMAC por tratar-se apenas de uma amostra de desempenho, tornando-se questionável a qualidade dos resultados. Com o intuito de confirmar o melhor desempenho, a última avaliação experimental consiste em 10 diferentes configurações de rede aleatoriamente construídas, cada uma com duração de 30 minutos. As configurações são montadas selecionando-se aleatoriamente tempos de transmissão e ociosidade para cada nó sem fio, repetindo-se esse processo até que o tempo total atinja a duração de cada experimento (30 minutos).

O resultado geral das configurações aleatórias é ilustrado pela figura 5.17. A figura 5.17a ilustra a taxa de trocas de cada solução enquanto a figura 5.17b mostra o desempenho geral. Novamente, o *regret* é normalizado em função da maior vazão. O maior número de trocas de FS-MAC resulta em pior desempenho, sendo esta a solução de menor otimalidade e maior *regret*. Esse resultado indica sucessivas escolhas ruins de FS-MAC em comparação aos demais. De fato, a otimalidade de FS-MAC é ligeiramente superior a 60%, apenas 10% melhor que o passeio aleatório (i.e. escolhas aleatórias). Por outro lado, AMAC é o mecanismo de seleção com menor número de trocas. Ainda assim, o desempenho de AMAC é inferior ao desempenho de SOMAC, indicando um comportamento conservador de não realizar trocas mesmo quando elas são desejáveis. A otimalidade de AMAC é inferior a 70% enquanto SOMAC possui otimalidade média de

aproximadamente 80%. Adicionalmente, SOMAC também possui melhor desempenho em termos de *regret* médio, embora esse seja similar ao valor de AMAC. A similaridade está relacionada à curta duração de cada experimento, onde nem CSMA/CA nem TDMA se sobrepõem um ao outro por muito tempo.

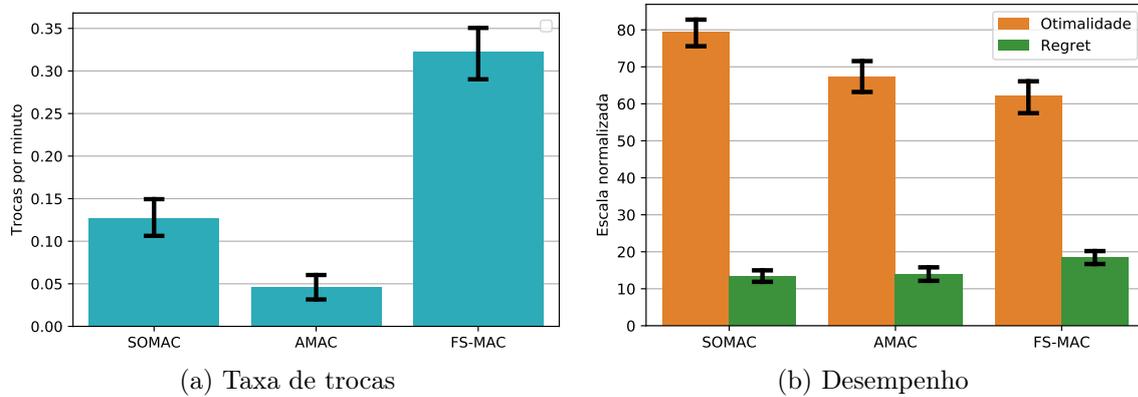


Figura 5.17: Comparação com o estado da arte em configurações aleatórias

### 5.3 Resumo

Este capítulo avalia o desempenho de SOMAC em redes sem fio reais empregando rádios definidos por software. As primeiras seções apresentam a metodologia utilizada para obtenção dos hiperparâmetros, tanto do algoritmo Q-Learning quanto das estratégias de exploração. A estratégia de exploração softmax é utilizada durante a avaliação experimental de SOMAC por indicativos de melhor desempenho que as demais.

Os resultados experimentais sugerem que SOMAC é capaz de operar com o melhor protocolo MAC em pelo menos 80% do tempo. Esse resultado é válido para diferentes métricas de desempenho de rede, utilizadas para o cálculo da recompensa do algoritmo Q-Learning. Além disso, SOMAC é até 25% melhor que seus competidores em termos de eficácia de seleção de protocolos MAC (i.e. otimalidade). O melhor desempenho também é visto em avaliações com cenários empregando momentos de transmissão aleatórios, onde SOMAC é superior ao estado da arte.

# Capítulo 6

## Conclusões

Este trabalho apresenta um sistema de troca de protocolos MAC para redes sem fio denominado SOMAC. Protocolos MAC são usualmente específicos à cenários de aplicação. Alguns protocolos possuem melhor desempenho em determinadas configurações de rede (e.g. redes densas) enquanto outros operam melhor em configurações distintas (e.g. redes esparsas). Uma vez que redes sem fio são dinâmicas, um único protocolo MAC é dificilmente o melhor o tempo todo. Assim, a solução apresentada nesta dissertação lida com esse problema. SOMAC seleciona o protocolo MAC de acordo com o contexto da rede sem fio, selecionando-se, ao longo do tempo, o protocolo que melhor se adéque às condições da rede.

Ao contrário de outros trabalhos encontrados na literatura, SOMAC lida com a dinamicidade de redes sem fio. O mecanismo de seleção de protocolos é capaz de adaptar-se à mudanças da rede, operando em ambientes não estacionários como redes sem fio. Além disso, SOMAC não depende de um especialista para ajustes no mecanismo de seleção como em outros trabalhos, dependentes de formulação matemática para novos protocolos ou definição de regras de inferência por exemplo.

O mecanismo de seleção de SOMAC é a maior contribuição desta dissertação. A seleção de protocolos é baseada no algoritmo de aprendizado por reforço Q-Learning. Esse algoritmo não apenas lida com ambientes não estacionários como também é independente de modelo, beneficiando sua aplicação no contexto de redes sem fio, cuja dinamicidade e complexidade dificultam a utilização de técnicas de aprendizado supervisionado ou baseadas em modelos. Os estados de Q-Learning são representados pelos protocolos MAC disponíveis em SOMAC ao passo que as ações consistem em manter ou alterar o protocolo MAC em uso. A ação tomada visa a otimização de uma métrica de desempenho  $m$ , podendo ser, por exemplo, uma métrica de desempenho de rede tal como vazão ou latência, ou até mesmo uma função de múltiplas métricas de rede.

Adicionalmente, SOMAC é implementado utilizando-se rádios definidos por software. A implementação é baseada na plataforma de processamento de sinais GNU Radio. Essa abordagem permite a avaliação de SOMAC em redes sem fio reais. Os resultados extraídos nesta dissertação consistem em avaliações experimentais de SOMAC no *testbed* UFMG.

Os resultados obtidos indicam um bom desempenho de SOMAC em diferentes situações. SOMAC é capaz de operar com o melhor protocolo MAC em, pelo menos, 80% do tempo, tentando a trocar o protocolo MAC quando necessário. Esse resultado se mostra verdadeiro mesmo em avaliações de configuração de rede aleatória, situação análoga à operação de uma rede sem fio no mundo real. Os resultados favoráveis se repetem para diferentes métricas de rede (i.e. como métrica de seleção  $m$ ), sendo avaliadas ambas a vazão e a latência média. Adicionalmente, SOMAC também apresenta melhor desempenho em comparação aos seus competidores em todos os casos avaliados. O ganho de SOMAC diante do estado da arte varia de 13% a 25% em termos de tempo de operação com o melhor protocolo (i.e. otimalidade).

Além disso, registramos aqui alguns trabalhos futuros passíveis de investigação a partir desta proposta. O primeiro é o impacto da troca de protocolos vista da experiência do usuário. Apesar de avaliarmos o desempenho do sistema em situações onde a troca ocorre ao longo do tempo, não investigamos a percepção do usuário sobre esse processo, podendo, eventualmente, acarretar em pausas (ou interrupções) de serviços de *streaming* de dados por exemplo.

Uma segunda proposta de investigação é a influência de múltiplos nós coordenadores de SOMAC em uma mesma região (i.e. MARL – *Multi-Agent Reinforcement Learning*). A presença de múltiplos agentes tomando decisões sobre o mesmo ambiente pode levar a instabilidade do sistema. Isso porque a escolha de um determinado protocolo MAC pode influenciar no meio como um todo, alterando-se, por exemplo, os níveis de interferência entre as redes sem fio e, conseqüentemente, o desempenho de cada uma. Assim, os múltiplos agentes podem desencadear uma sequência de múltiplas trocas de protocolos como resposta às ações tomadas por outros agentes, dificultando a convergência do sistema.

Por fim, a terceira proposta de trabalhos futuros consiste em explorar o uso de técnicas de atualização de modelos de aprendizado de máquina para lidar com a natureza não estacionária de redes sem fio. Apesar dos problemas de esquecimento catastrófico e estabilidade-plasticidade, modelos de aprendizado de máquina adaptáveis ao longo do tempo (também conhecidos como aprendizado *online*) podem ser uma alternativa ao uso de aprendizado por reforço, sendo interessante uma comparação de desempenho. Técnicas de atualização de modelos também possibilitam a utilização de algoritmos

de aprendizado profundo por reforço (i.e. *Deep Reinforcement Learning*), permitindo uma modelagem mais minuciosa do problema. Por exemplo, os estados de Q-Learning podem ser modelados considerando-se diferentes níveis de interferência uma vez que um mesmo protocolo MAC possivelmente terá desempenho diferente para diferentes níveis de interferência. Essa abordagem não escala bem na formulação tradicional do algoritmo Q-Learning, onde estados e ações são representados de forma tabular. Isso porque o nível de interferência é uma variável contínua, resultando em um espaço infinito de estados. Além disso, várias outras variáveis podem ser levadas em consideração para melhor representação da rede sem fio, o que dificulta mais ainda a escalabilidade. Nesses casos, técnicas de aprendizado profundo por reforço possibilitam a aproximação dos estados através de redes neurais, tornando interessante a investigação de formas de mantê-las atualizadas em ambientes não estacionários (caso de SOMAC).



# Referências Bibliográficas

- Al-mefleh, H. (2009). *Design and analysis of MAC protocols for wireless networks*. Ph. D. thesis, Iowa State University.
- Bicheno, S. (2017). Vodafone explores software-defined radio with a squeeze of lime. <http://telecoms.com/483345/vodafone-explores-software-defined-radio-with-a-squeeze-of-lime/>.
- Bifet, A., G. Holmes, R. Kirkby, and B. Pfahringer (2010, August). MOA: Massive Online Analysis. *J. Mach. Learn. Res.* 11, 1601–1604.
- Bloessl, B., A. Puschmann, C. Sommer, and F. Dressler (2014). Timings matter: standard compliant ieee 802.11 channel access for a fully software-based SDR architecture. In *Proceedings of the 9th ACM international workshop on Wireless network test-beds, experimental evaluation and characterization - WiNTECH '14*, Maui, Hawaii, USA, pp. 57–64. ACM Press.
- Bloessl, B., M. Segata, C. Sommer, and F. Dressler (2013). An IEEE 802.11a/g/p OFDM receiver for GNU radio. In *Proceedings of the second workshop on Software radio implementation forum - SRIF '13*, Hong Kong, China, pp. 9. ACM Press.
- Boutaba, R., M. A. Salahuddin, N. Limam, S. Ayoubi, N. Shahriar, F. Estrada-Solano, and O. M. Caicedo (2018, Jun). A comprehensive survey on machine learning for networking: evolution, applications and research opportunities. *Journal of Internet Services and Applications* 9(1), 16.
- Busch, C., M. Magdon-Ismail, F. Sivrikaya, and B. Yener (2004). Contention-Free MAC Protocols for Wireless Sensor Networks. In D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, and R. Guerraoui (Eds.), *Distributed Computing*, Volume 3274, pp. 245–259. Berlin, Heidelberg: Springer Berlin Heidelberg.

- Cesa-Bianchi, N. and G. Lugosi (2006). *Prediction, learning, and games*. Cambridge; New York: Cambridge University Press. OCLC: 70056026.
- Choudhury, R., Xue Yang, R. Ramanathan, and N. Vaidya (2006, May). On designing MAC protocols for wireless networks using directional antennas. *IEEE Transactions on Mobile Computing* 5(5), 477–491.
- Cordeiro, J. R. S., D. F. Macedo, and L. F. M. Vieira (2018, April). FS-MAC: A flexible MAC platform for wireless networks. In *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6.
- da Silva, W. S., J. R. S. Cordeiro, D. F. Macedo, M. A. M. Vieira, L. F. M. Vieira, and J. M. S. Nogueira (2015). Introdução a rádios definidos por software com aplicações em GNU radio. Minicurso apresentado no XXXIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos.
- Demirkol, I., C. Ersoy, and F. Alagoz (2006, April). MAC protocols for wireless sensor networks: a survey. *IEEE Communications Magazine* 44(4), 115–121.
- Ettus Research (2018a). Products. <https://www.ettus.com/product>.
- Ettus Research (2018b). Usrp hardware driver and usrp manual. <https://files.ettus.com/manual/index.html>.
- Ettus Research (nd). RFNoC™ (RF Network on Chip). <https://www.ettus.com/sdr-software/detail/rf-network-on-chip>.
- Evans, D. (2011, apr). The internet of things: How the next evolution of the internet is changing everything. Technical report, CiscoIBSG.
- Ferrand, P., M. Amara, S. Valentin, and M. Guillaud (2016, July). Trends and challenges in wireless channel modeling for evolving radio access. *IEEE Communications Magazine* 54(7), 93–99.
- Forum, W. I. (2011). Software defined radio - rate of adoption. [http://www.wirelessinnovation.org/sdr\\_rate\\_of\\_adoption](http://www.wirelessinnovation.org/sdr_rate_of_adoption).
- FUTEBOL (2018). Testbeds. <http://www.ict-futebol.org.br/index.php/infrastructure/testbeds/>.
- Garivier, A. and E. Moulines (2011). On upper-confidence bound policies for switching bandit problems. In J. Kivinen, C. Szepesvári, E. Ukkonen, and T. Zeugmann (Eds.),

- Algorithmic Learning Theory*, Berlin, Heidelberg, pp. 174–188. Springer Berlin Heidelberg.
- Gepperth, A. and B. Hammer (2016). Incremental learning algorithms and applications. In *European Symposium on Artificial Neural Networks (ESANN)*.
- GNURadio (2017). Guided tutorial programming topics. [https://wiki.gnuradio.org/index.php/Guided\\_Tutorial\\_Programming\\_Topics](https://wiki.gnuradio.org/index.php/Guided_Tutorial_Programming_Topics).
- GNURadio (2018). About gnu radio. <https://www.gnuradio.org/>.
- Gonçalves, A. V. G. S., D. F. Macedo, and L. F. M. Vieira (2018, May). FS-MAC+: Troca flexível do protocolo de acesso ao meio em redes sem fio empregando aprendizado de máquina. In *Anais do Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*, Volume 36.
- Great Scott Gadgets (2018). Hackrf one. <https://greatscottgadgets.com>.
- Guo, W., L. Huang, L. Chen, H. Xu, and J. Xie (2012, October). An Adaptive Collision-Free MAC protocol based on TDMA for Inter-Vehicular communication. In *2012 International Conference on Wireless Communications and Signal Processing (WCSP)*, Huangshan, China, pp. 1–6. IEEE.
- Hu, W., H. Yousefi'zadeh, and X. Li (2011, November). Load Adaptive MAC: A Hybrid MAC Protocol for MIMO SDR MANETs. *IEEE Transactions on Wireless Communications* 10(11), 3924–3933.
- Huang, K.-C., X. Jing, and D. Raychaudhuri (2009, August). MAC Protocol Adaptation in Cognitive Radio Networks: An Experimental Study. In *2009 Proceedings of 18th International Conference on Computer Communications and Networks*, San Francisco, CA, USA, pp. 1–6. IEEE.
- IEEE (2016). IEEE Std 802.11<sup>TM</sup>-2016, IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 11: Wireless LAN Medium Access Control. Technical report, IEEE.
- ITU-R (1997). M.1073-1 - Digital cellular land mobile telecommunication systems. Technical report, ITU-R.
- John, G. H. (1998). When the Best Move Isn't Optimal: Q-learning with Exploration. Technical report, Stanford.

- Koenig, S. and R. G. Simmons (1992). Complexity Analysis of Real-Time Reinforcement Learning. Technical report, CARNEGIE-MELLON UNIV PITTSBURGH PA SCHOOL OF COMPUTER SCIENCE.
- Kuleshov, V. and D. Precup (2014, February). Algorithms for multi-armed bandit problems. Technical report, CoRR. arXiv: 1402.6028.
- Kurose, J. F. and K. W. Ross (2012). *Computer Networking: A Top-Down Approach (6th Edition)* (6th ed.). Pearson.
- Libin Jiang and J. Walrand (2010, June). A Distributed CSMA Algorithm for Throughput and Utility Maximization in Wireless Networks. *IEEE/ACM Transactions on Networking* 18(3), 960–972.
- Limited, S. R. S. (2016). About us. <http://www.softwareradiosystems.com/about-us/>.
- Lobo, A. P. (2017). Brasil possui 280 milhões de dispositivos móveis conectáveis à Internet.
- Marinescu, A. (2016). *Prediction-Based Multi-Agent Reinforcement Learning for Inherently Non-Stationary Environments*. Ph. D. thesis, University of Dublin, Trinity College.
- Mitola, J. (1993). Software Radios Survey, Critical Evaluation and Future Directions. *IEEE Aerospace and Electronic Systems Magazine* 8(4), 25–36.
- Mitola, J. (2000, May). *Cognitive Radio — An Integrated Agent Architecture for Software Defined Radio*. DTech thesis, Royal Institute of Technology (KTH), Kista, Sweden.
- Mnih, V., K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis (2015, February). Human-level control through deep reinforcement learning. *Nature* 518(7540), 529–533.
- Moura, H., G. V. C. Bessa, M. A. M. Vieira, and D. F. Macedo (2015, May). Ethanol: Software defined networking for 802.11 Wireless Networks. In *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, Ottawa, ON, Canada, pp. 388–396. IEEE.

- Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. The MIT Press.
- Nohrborg, M. (2018). Self-Organizing Networks. <http://www.3gpp.org/technologies/keywords-acronyms/105-son>.
- Qiao, M., H. Zhao, S. Wang, and J. Wei (2016). MAC Protocol Selection Based on Machine Learning in Cognitive Radio Networks. In *Wireless Personal Multimedia Communications (WPMC), 2016 19th International Symposium on*, pp. 453–458. IEEE.
- Rhee, I., A. Warriar, M. Aia, J. Min, and M. L. Sichitiu (2008, June). Z-MAC: A Hybrid MAC for Wireless Sensor Networks. *IEEE/ACM Transactions on Networking* 16(3), 511–524.
- RTL-SDR (2018). About rtl-sdr. <https://www.rtl-sdr.com/about-rtl-sdr/>.
- Schmid, T., O. Sekkat, and M. B. Srivastava (2007). An experimental study of network performance impact of increased latency in software defined radios. In *Proceedings of the the second ACM international workshop on Wireless network testbeds, experimental evaluation and characterization - WinTECH '07*, Montreal, Quebec, Canada, pp. 59. ACM Press.
- Sutton, R. S. and A. G. Barto (1998). *Reinforcement learning: an introduction*. Adaptive computation and machine learning. Cambridge, Mass: MIT Press.
- Systems, C. S. (2018). Software defined radio satellite transceiver links. <https://www.clutchspace.com/software-defined-radio-satellite-li>.
- Tan, K., H. Liu, J. Zhang, Y. Zhang, J. Fang, and G. M. Voelker (2011, January). Sora: high-performance software radio using general-purpose multi-core processors. *Communications of the ACM* 54(1), 99.
- The Sora Core Team (2012). *The Sora manual*. The Microsoft Research Software Radio.
- Tijmsma, A. D., M. M. Drugan, and M. A. Wiering (2016, December). Comparing exploration strategies for Q-learning in random stochastic mazes. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, Athens, Greece, pp. 1–8. IEEE.
- Tsividis, P., T. Pouncy, J. L. Xu, J. B. Tenenbaum, and S. J. Gershman (2017). Human learning in atari. In *AAAI Spring Symposia*.

- Vodafone (2018). Vodafone announces successful trials of new technologies that will bring cost-effective connectivity to millions of people in Africa and India. <https://www.vodafone.com/content/index/what/technology-blog/cost-effective-connectivity-in-africa-and-india.html>.
- Watkins, C. J. C. H. (1989). *Learning from Delayed Rewards*. Ph. D. thesis, King's College, Cambridge, UK.
- Yang, X., L. Wang, J. Su, and Y. Gong (2018, June). Hybrid MAC Protocol Design for Mobile Wireless Sensors Networks. *IEEE Sensors Letters* 2(2), 1–4.
- Ye, Q., W. Zhuang, L. Li, and P. Vigneron (2016, November). Traffic-Load-Adaptive Medium Access Control for Fully Connected Mobile Ad Hoc Networks. *IEEE Transactions on Vehicular Technology* 65(11), 9358–9371.
- Yu, Y., T. Wang, and S. C. Liew (2017, November). Deep-Reinforcement Learning Multiple Access for Heterogeneous Wireless Networks. In *2018 IEEE International Conference on Communications (ICC)*, pp. 1–7. IEEE. arXiv: 1712.00162.
- Zhou, A., B. Wang, X. Sun, X. You, H. Sun, and T. Li (2015, February). SLQE: An Improved Link Quality Estimation based on Four-bit LQE. *International Journal of Future Generation Communication and Networking* 8(1), 149–160.
- Ziouva, E. and T. Antonakopoulos (2002). CSMA/CA performance under high traffic conditions: throughput and delay analysis. *Computer Communications* 25(3), 313–321.