

Uma abordagem Q-Learning para escalonamento dinâmico de comunicação do TSCH

Victor S. Cardel¹

Bruno P. Santos¹

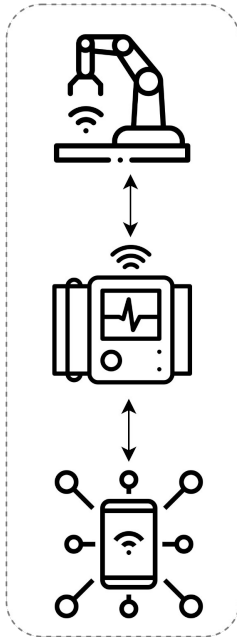
Paulo H. Rettore²

¹Instituto de Ciência da Computação, Universidade Federal da Bahia (UFBA)

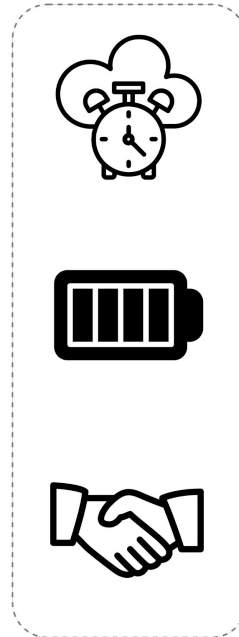
²Communication Systems Department, Fraunhofer FKIE

Introdução

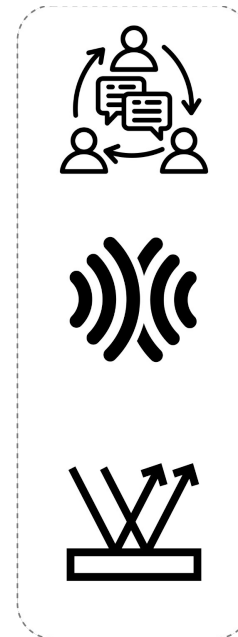
Indústria 4.0



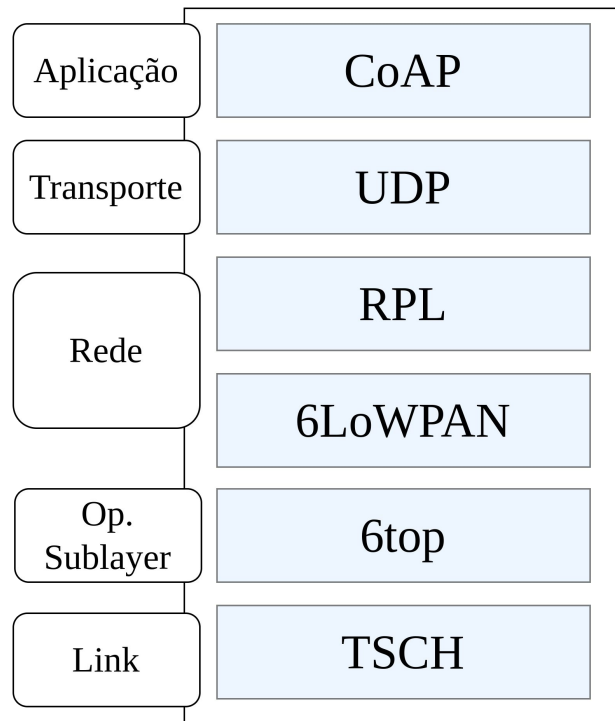
Requisitos



Desafios

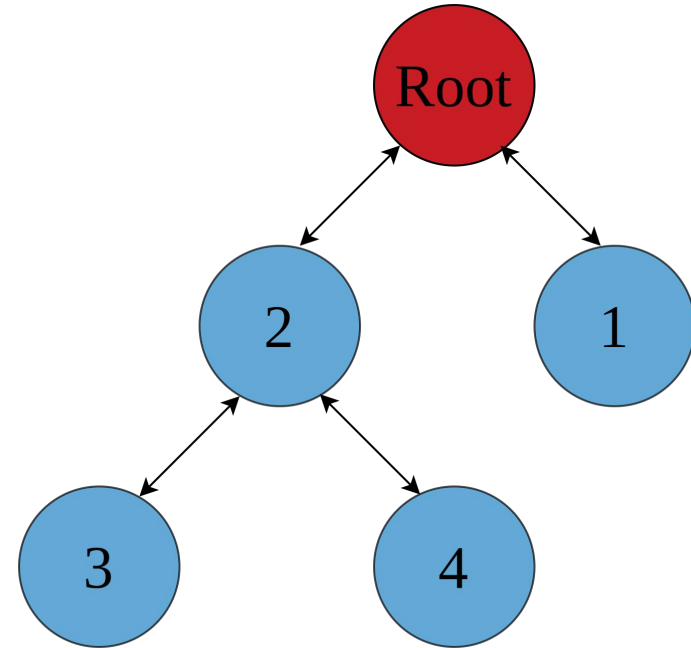
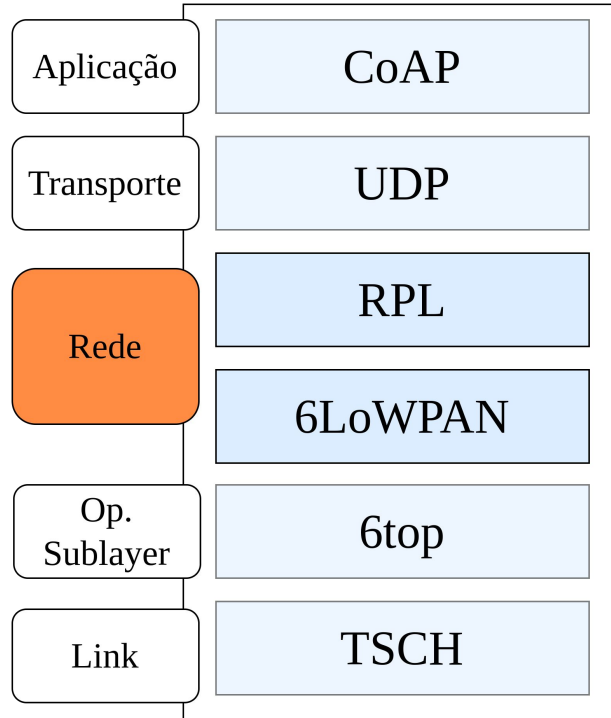


Arquitetura 6TiSCH

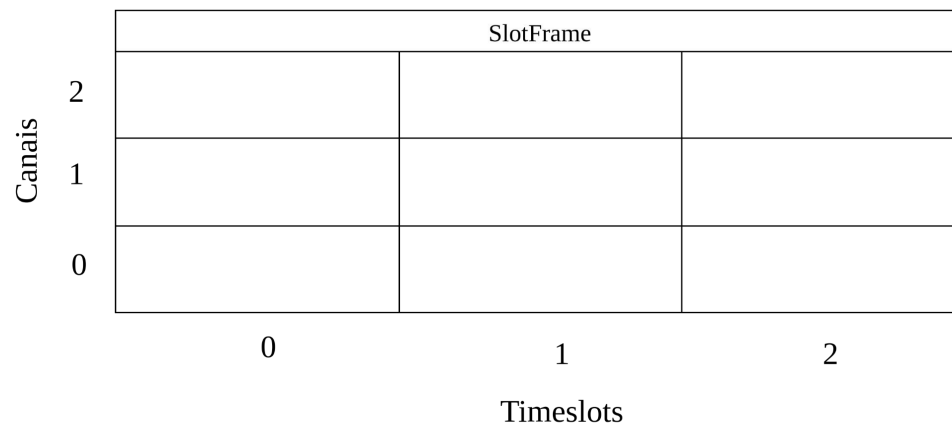
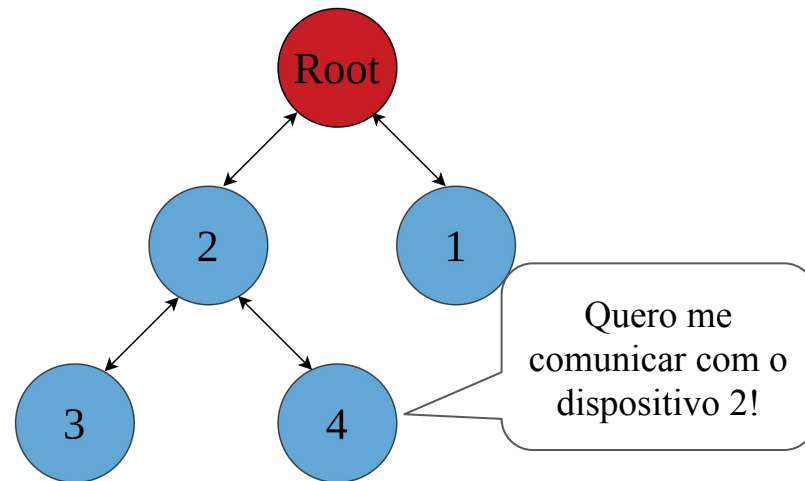
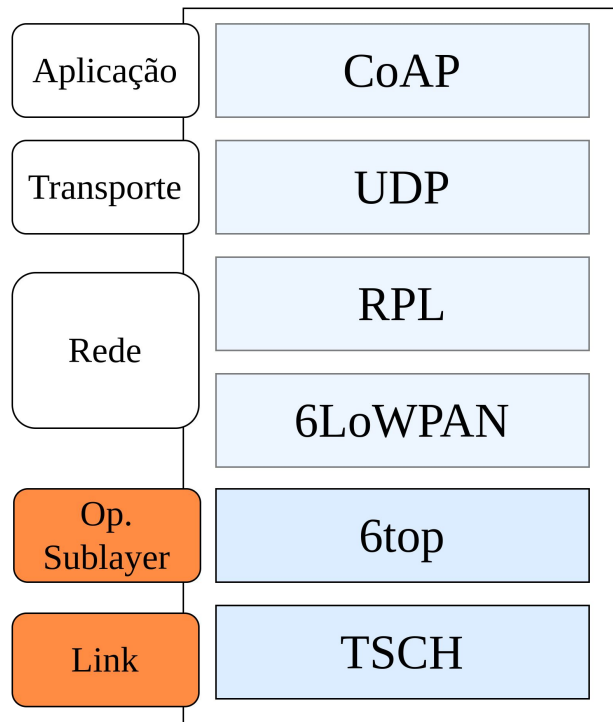


- **Arquitetura para redes sem fio de baixa potência.**
- Possui a proposta de prover **baixo consumo de energia com alta confiabilidade.**
- Cumpre o papel de **conectar redes sem fio de baixa potência com a internet** através do IPV6.

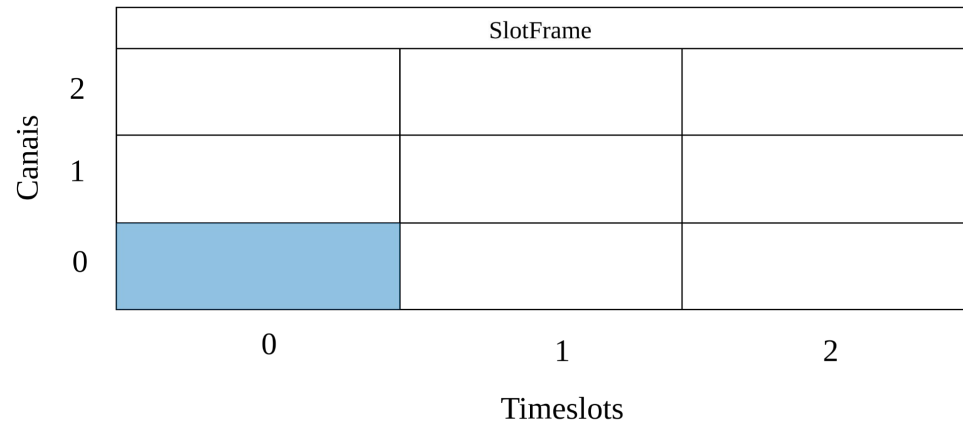
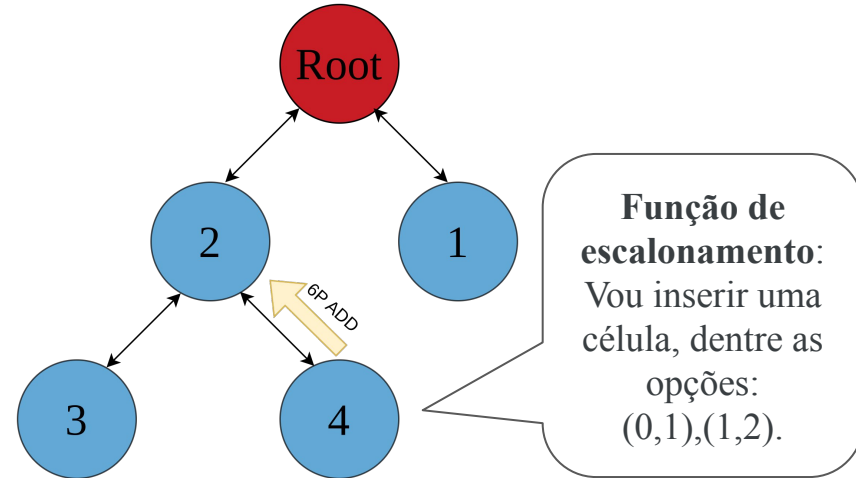
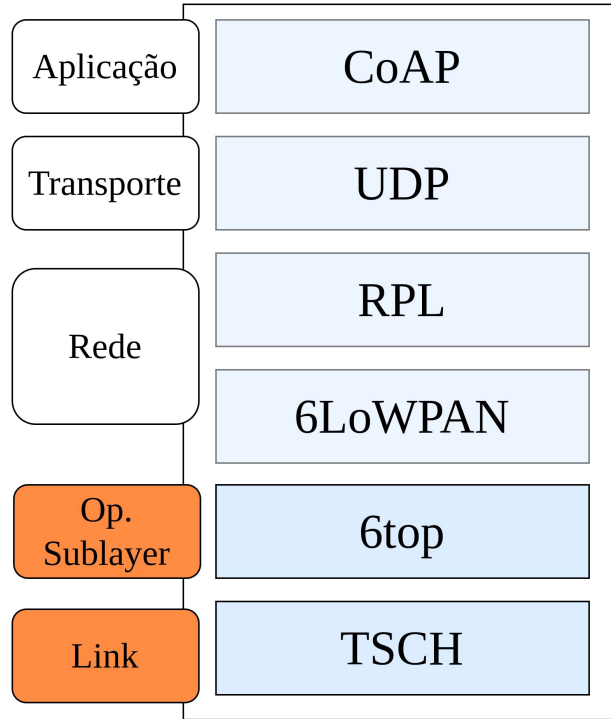
Arquitetura 6TiSCH



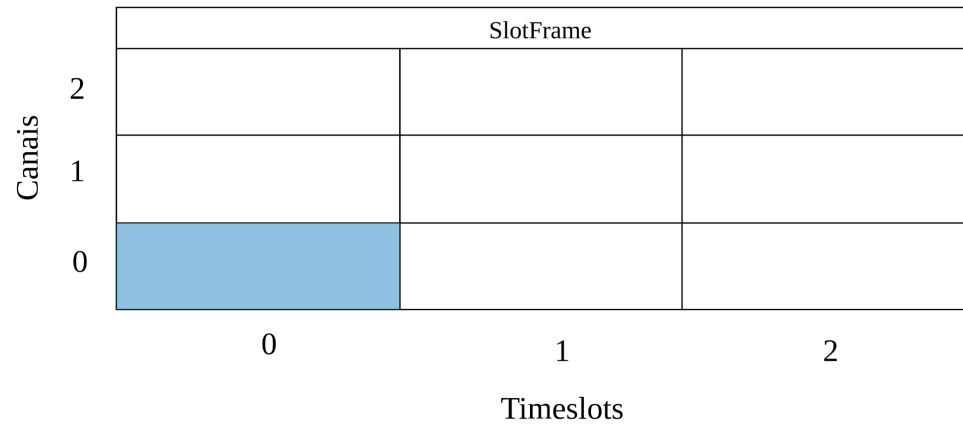
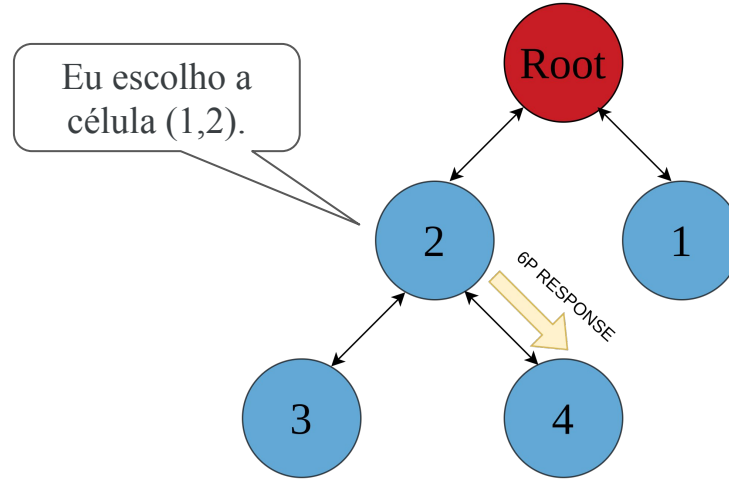
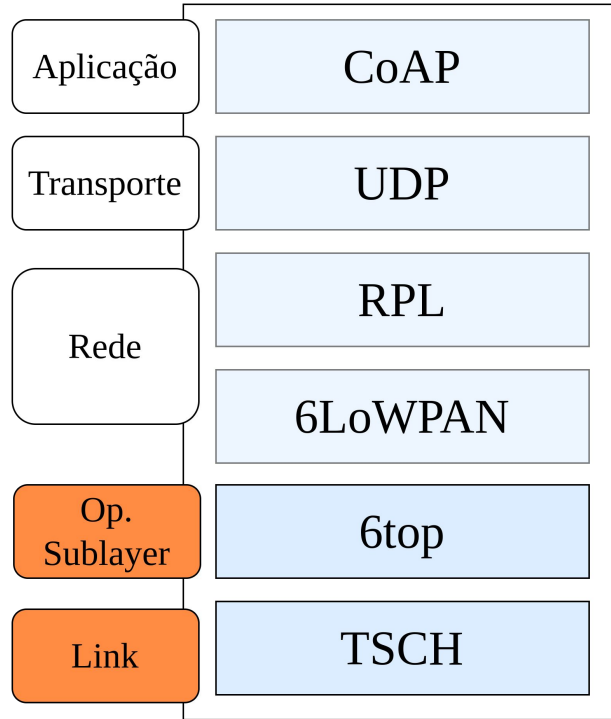
Arquitetura 6TiSCH



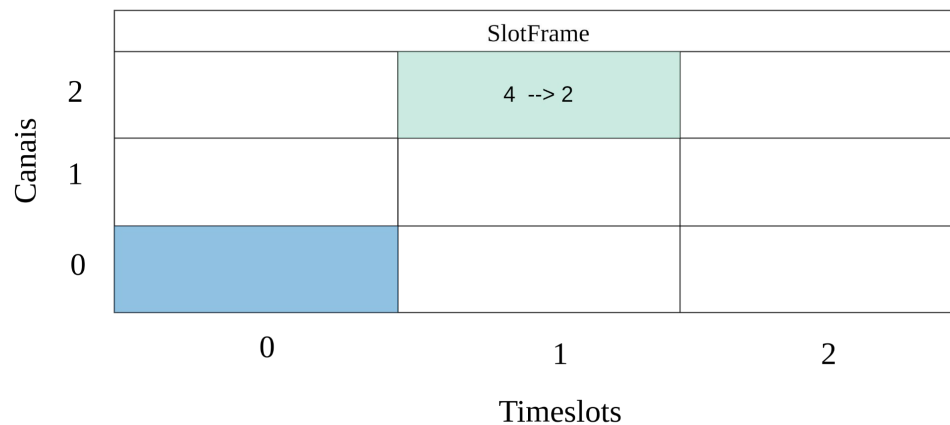
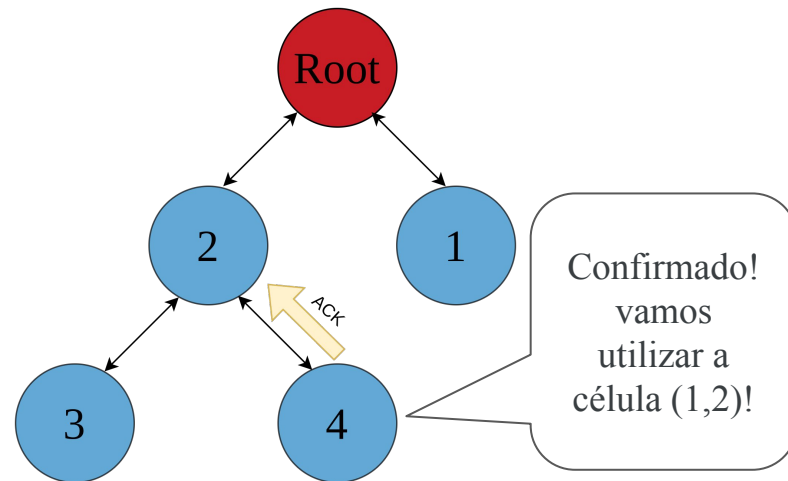
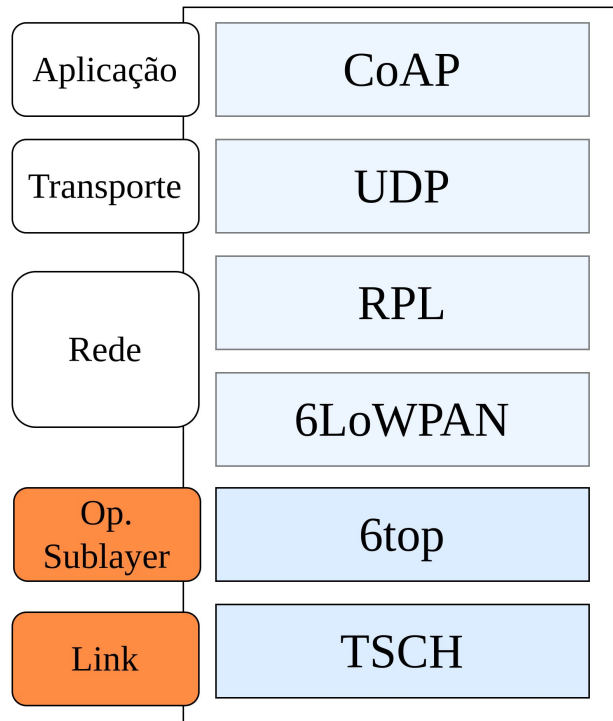
Arquitetura 6TiSCH



Arquitetura 6TiSCH



Arquitetura 6TiSCH



Contribuições

- ✓ **Q-Learning para escalonamento dinâmico de comunicação TSCH.**
- ✓ **Um comparativo qualitativo entre as abordagens.**
- ✓ **Resultados experimentais.**
- ✓ **Metodologia de código aberto.**

Trabalhos Relacionados

Referências	Estratégias	Otimização	Q-Learning	Tráfego	Buffer	Energia
Hamza and Kaddoum 2019	Previsão do tráfego	Escalonamento	✗	✓	✗	✗
Hauweele et al. 2020	Adaptação ao tráfego	Escalonamento	✗	✓	✗	✗
Domingo-Prieto et al. 2016	Controlador PID	Escalonamento	✗	✓	✓	✗
Palatella et al. 2012	Coloração/ Emparelhamento	Escalonamento	✗	✗	✗	✗
Fawwaz and Chung 2023	Q-Learning	Envio/Supressão DIOs	✓	✗	✓	✗
Nguyen-Duy et al. 2019	Q-Learning	Escalonamento	✓	✗	✓	✓
Bekar et al. 2023	Q-Learning	OFs para RPL	✓	✓	✓	✗
Pratama and Chung 2022	Q-Learning	Escalonamento	✓	✗	✓	✗
Este Trabalho	Q-Learning	Escalonamento	✓	✓	✓	✓

Trabalhos Relacionados

Referências	Estratégias	Otimização	Q-Learning	Tráfego	Buffer	Energia
Hamza and Kaddoum 2019	Previsão do tráfego	Escalonamento	✗	✓	✗	✗
Hauweele et al. 2020	Adaptação ao tráfego	Escalonamento	✗	✓	✗	✗
Domingo-Prieto et al. 2016	Controlador PID	Escalonamento	✗	✓	✓	✗
Palatella et al. 2012	Coloração/ Emparelhamento	Escalonamento	✗	✗	✗	✗
Fawwaz and Chung 2023	Q-Learning	Envio/Supressão DIOs	✓	✗	✓	✗
Nguyen-Duy et al. 2019	Q-Learning	Escalonamento	✓	✗	✓	✓
Bekar et al. 2023	Q-Learning	OFs para RPL	✓	✓	✓	✗
Pratama and Chung 2022	Q-Learning	Escalonamento	✓	✗	✓	✗
Este Trabalho	Q-Learning	Escalonamento	✓	✓	✓	✓

Trabalhos Relacionados

Referências	Estratégias	Otimização	Q-Learning	Tráfego	Buffer	Energia
Hamza and Kaddoum 2019	Previsão do tráfego	Escalonamento	✗	✓	✗	✗
Hauweele et al. 2020	Adaptação ao tráfego	Escalonamento	✗	✓	✗	✗
Domingo-Prieto et al. 2016	Controlador PID	Escalonamento	✗	✓	✓	✗
Palatella et al. 2012	Coloração/ Emparelhamento	Escalonamento	✗	✗	✗	✗
Fawwaz and Chung 2023	Q-Learning	Envio/Supressão DIOs	✓	✗	✓	✗
Nguyen-Duy et al. 2019	Q-Learning	Escalonamento	✓	✗	✓	✓
Bekar et al. 2023	Q-Learning	OFs para RPL	✓	✓	✓	✗
Pratama and Chung 2022	Q-Learning	Escalonamento	✓	✗	✓	✗
Este Trabalho	Q-Learning	Escalonamento	✓	✓	✓	✓

Métricas

$$t_b = \left(\sum_{i=S-k}^S Q \right) / k$$



S = Slotframe atual

k = 10 Slotframes

Q = tamanho da fila

RxAck = pacotes recebidos
reconhecidos

$e_l =$

**Energia
remanescente**

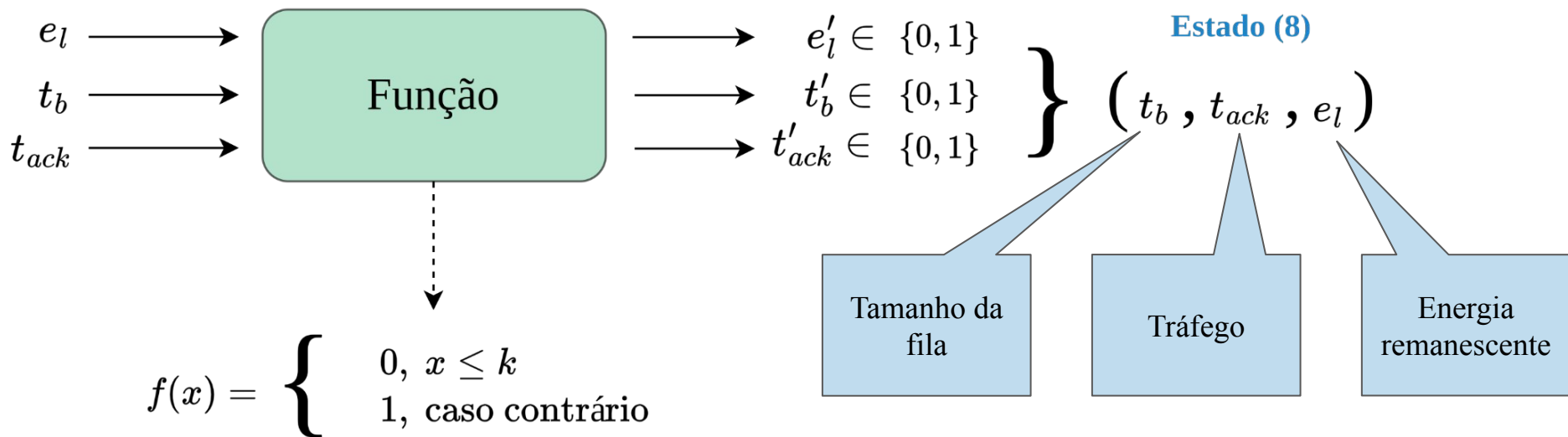


$$t_{ack} = \left(\sum_{i=S-k}^S RxAck \right) / k$$



¹[A realistic energy consumption model for TSCH networks](#)

Modelagem



Modelagem

Estados

$(0, 0, 0)$

$(0, 0, 1)$

$(0, 1, 0)$

...

$(1, 1, 1)$

Modelagem

Estados	Ações		
	A_1	A_2	A_3
$(0, 0, 0)$			
$(0, 0, 1)$			
$(0, 1, 0)$			
\dots			
$(1, 1, 1)$			

Q-Table

$A_1 =$ **Inserir célula**

$A_2 =$ **Remover célula**

$A_3 =$ **Mantém**

Modelagem

		Ações		
Estados		A_1	A_2	A_3
	$(0, 0, 0)$	$r_{1,1}$	$r_{1,2}$	$r_{1,3}$
s_f	$(0, 0, 1)$	$r_{2,1}$	$r_{2,2}$	$r_{1,3}$
	$(0, 1, 0)$...	
	...			
	$(1, 1, 1)$	$r_{8,1}$	$r_{8,2}$	$r_{8,3}$

Estado desejado

s_f

Recompensas esperadas

$A_1 =$ **Inserir célula**
 $A_2 =$ **Remove célula**
 $A_3 =$ **Mantém**

Q-Table

Modelagem

Legenda

Estado	s
Ação	a
Próximo estado	s_{next}
Função de recompensa	$r(s, a)$
Entrada da Q-table	$Q(s, a)$

Recompensa.

Equação de Bellman

$$\Delta Q(s, a) = r(s, a) + \gamma \times \max_a Q(s_{next}, a)$$

$$Q(s, a)_{new} = (1 - \alpha) \times Q(s, a)_{old} + \alpha \times \Delta Q(s, a)$$

Entrada nova.

Entrada antiga.

Modelagem

$$r(s, a) = \begin{cases} \beta, \text{ se } s = s_f \\ (1 - t'_b) + (1 - t'_{ack}) + e'_l \end{cases}$$

- Maior recompensa se atingir o estado desejado
- Caso contrário, recompensa é proporcional às métricas desejadas.

Modelagem

- A inserção ou remoção de uma célula exige a invocação do protocolo 6P.
- *Overhead* de mensagens de controle.
- Estratégia de inserir ou remover mais de uma célula por decisão.

Inserção:

$$C_i = t'_b + e'_l + t'_{ack}$$

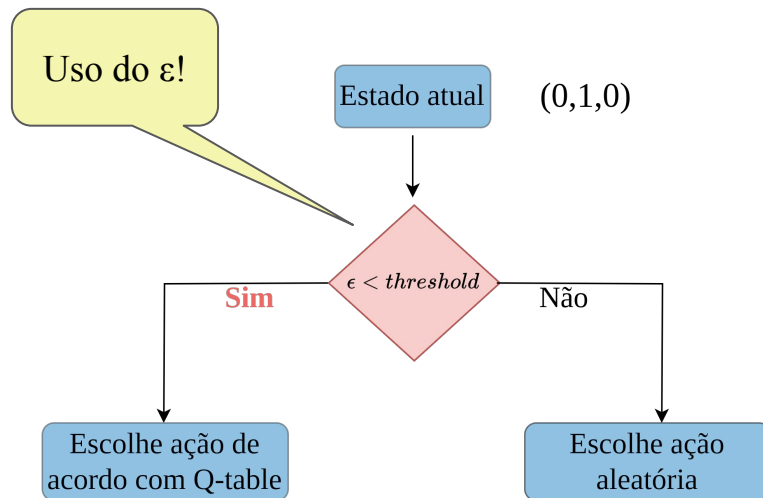
Remoção:

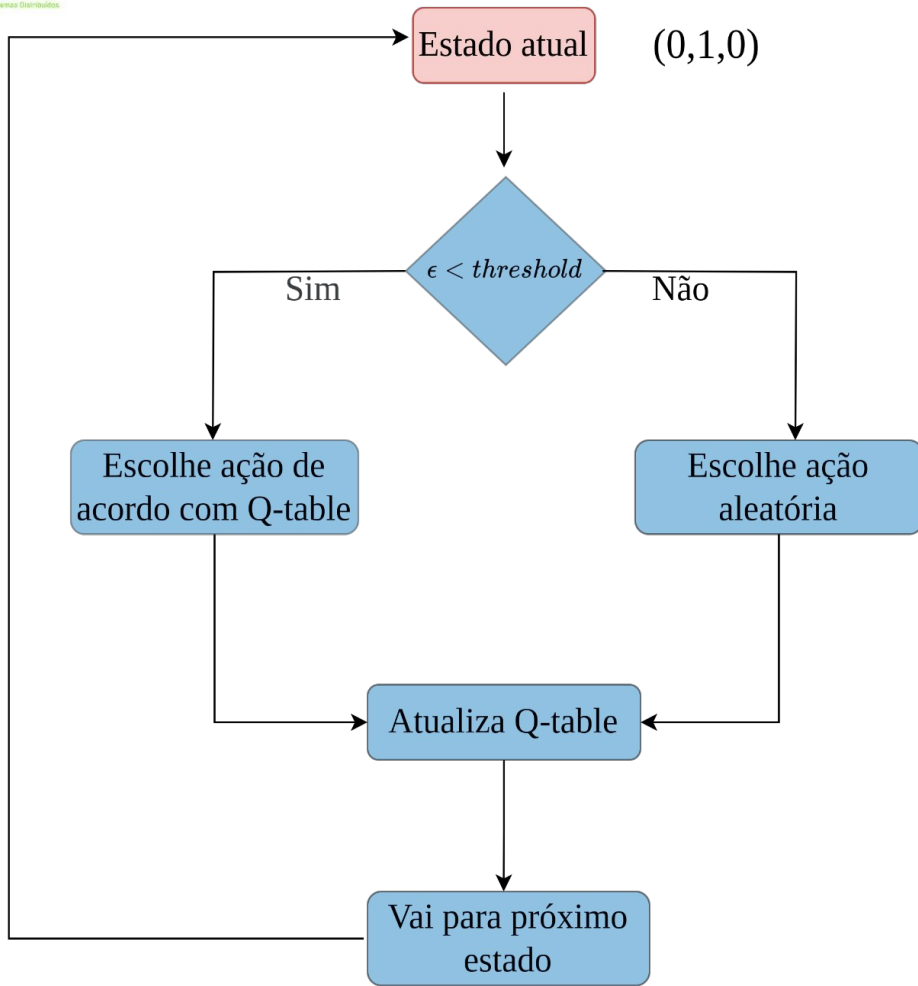
$$C_r = 3 - C_i$$

Modelagem

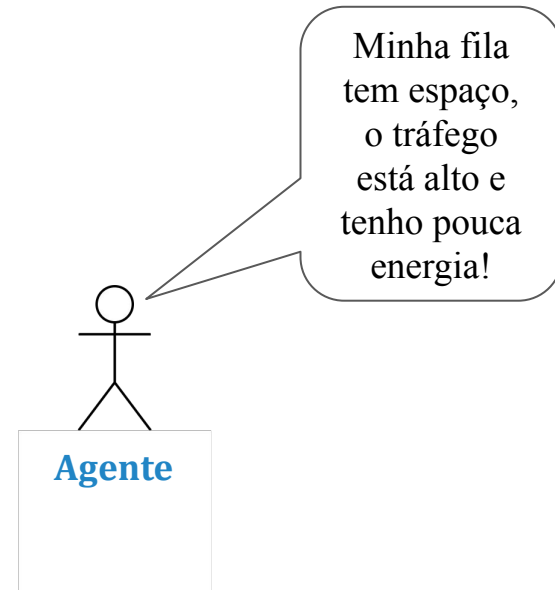
- O Q-Learning depende de um parâmetro ϵ
- Esse parâmetro governa a proporção entre *Exploration* e *Exploitation*.
- O valor de ϵ varia ao longo da execução do algoritmo segundo a fórmula:

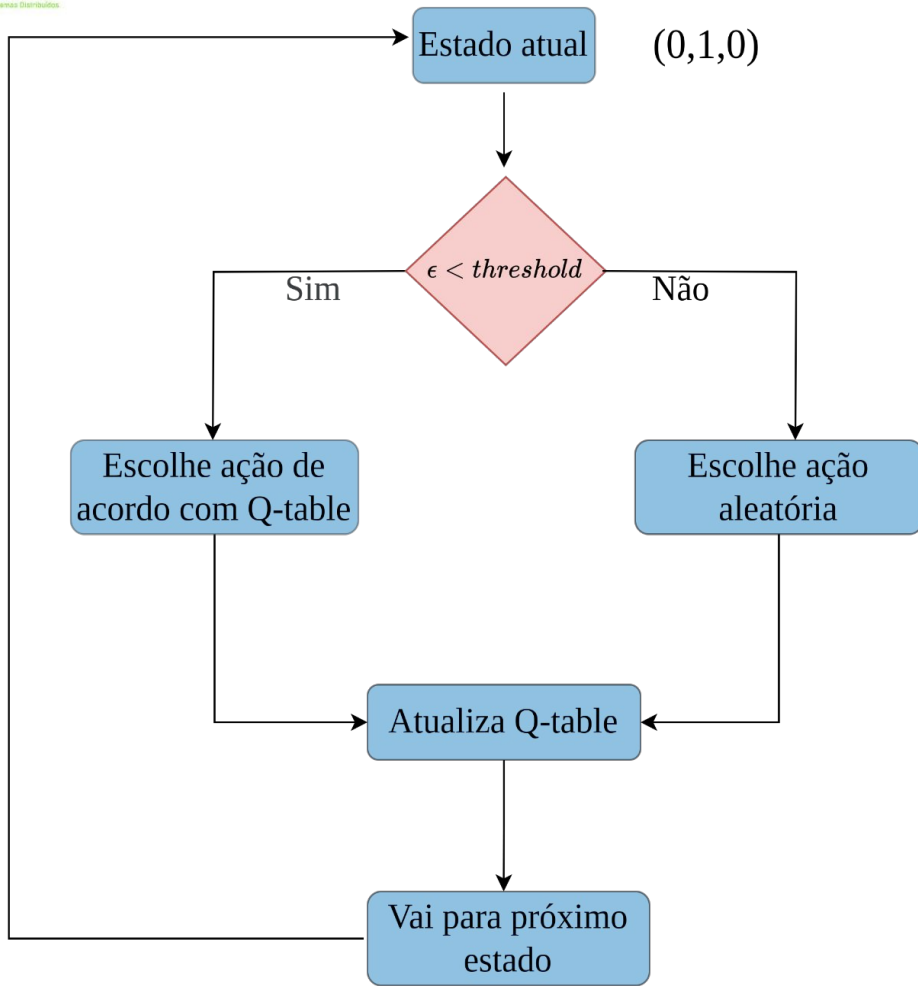
$$\epsilon = \epsilon_{min} + (\epsilon_{max} - \epsilon_{min}) \times e^{-decayRate \times episode}$$



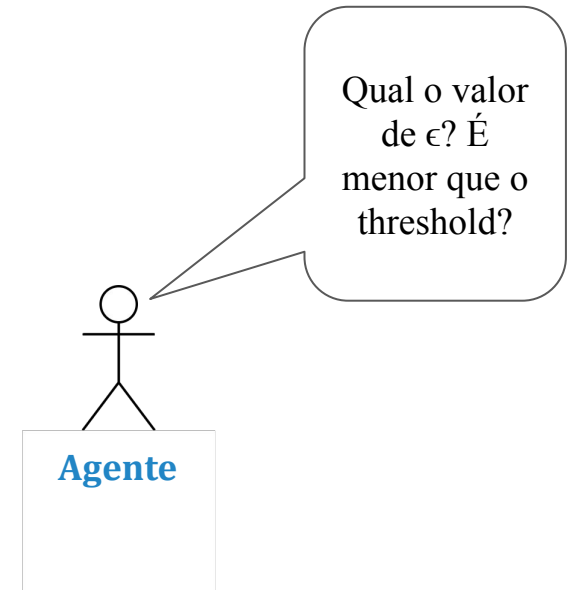


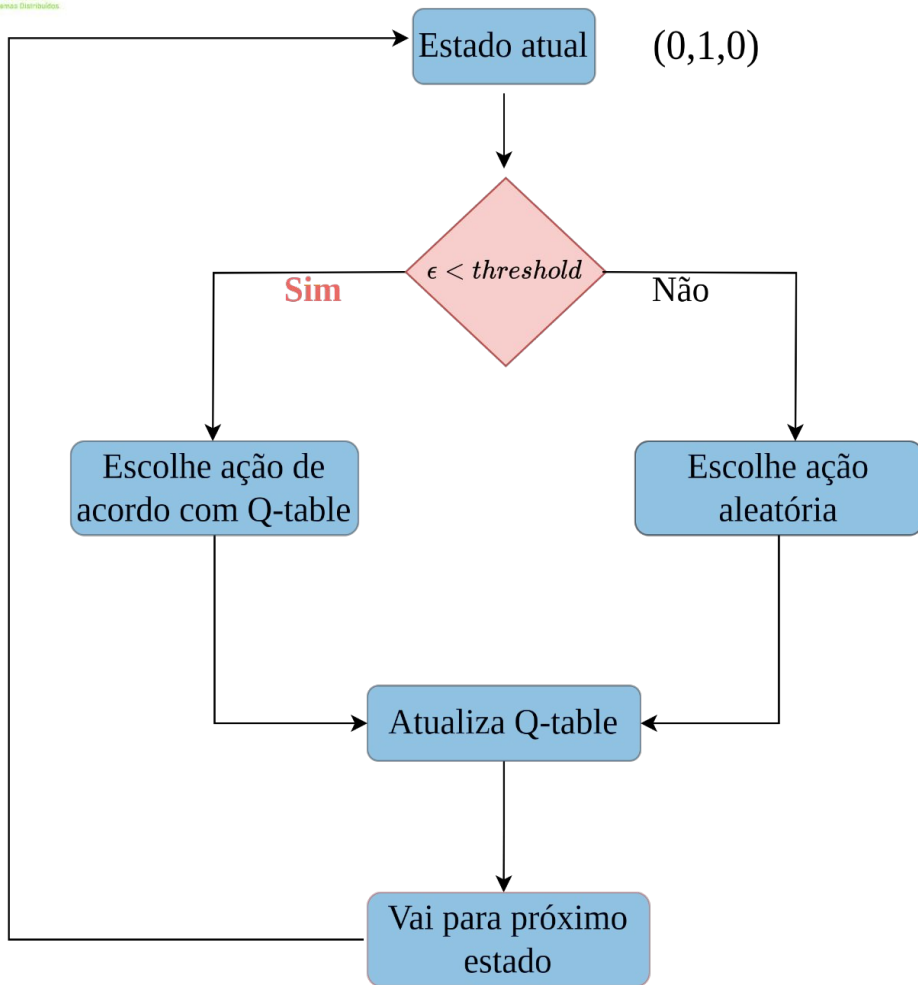
		Ações		
		A_1	A_2	A_3
Estados	$(0, 0, 0)$	$r_{1,1}$	$r_{1,2}$	$r_{1,3}$
	$s_f (0, 0, 1)$	$r_{2,1}$	$r_{2,2}$	$r_{1,3}$
	$(0, 1, 0)$	3	2	1
	
	$(1, 1, 1)$	$r_{8,1}$	$r_{8,2}$	$r_{8,3}$



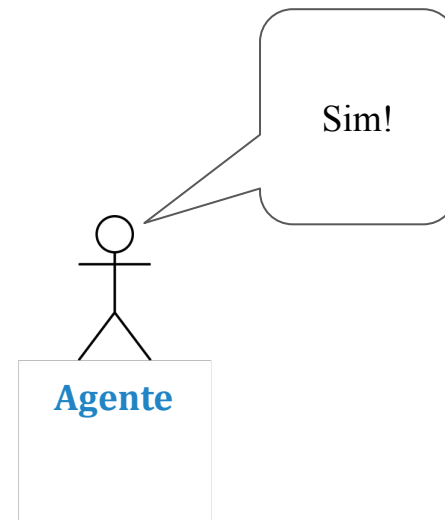


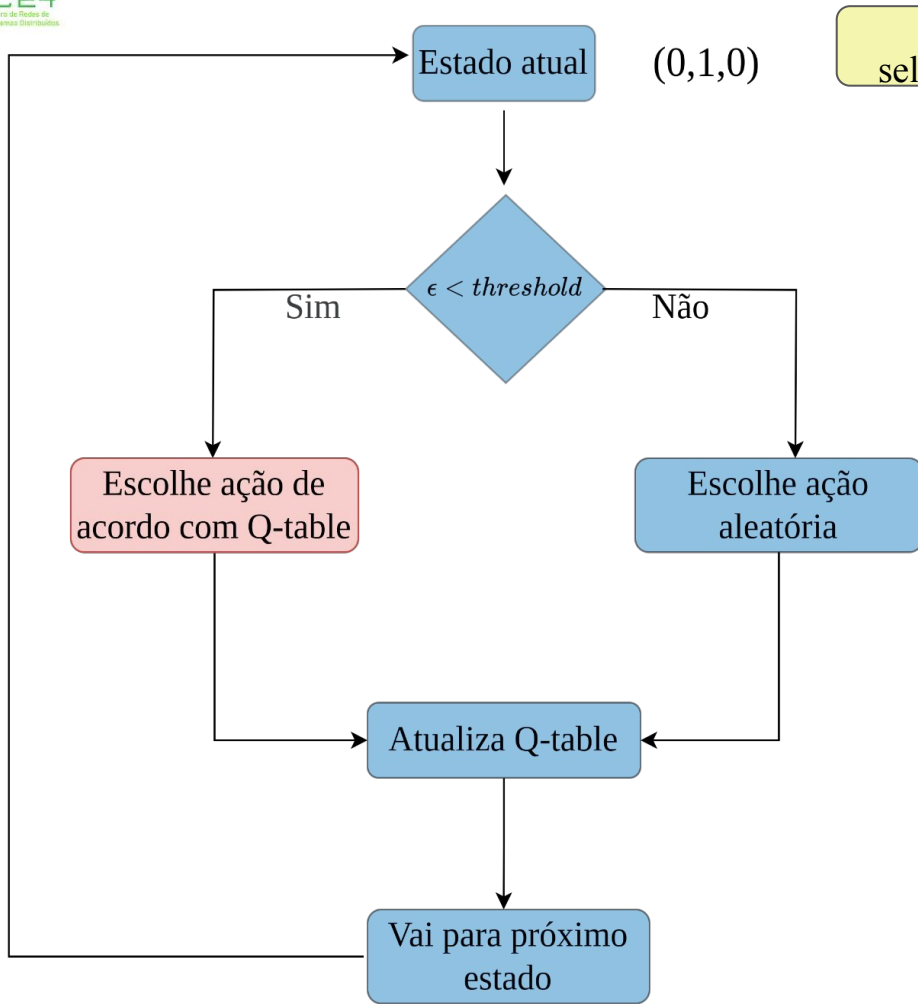
Estados	Ações		
	A_1	A_2	A_3
$(0, 0, 0)$	$r_{1,1}$	$r_{1,2}$	$r_{1,3}$
$s_f (0, 0, 1)$	$r_{2,1}$	$r_{2,2}$	$r_{1,3}$
$(0, 1, 0)$	3	2	1
...		...	
$(1, 1, 1)$	$r_{8,1}$	$r_{8,2}$	$r_{8,3}$





Estados	Ações		
	A_1	A_2	A_3
$(0, 0, 0)$	$r_{1,1}$	$r_{1,2}$	$r_{1,3}$
$s_f (0, 0, 1)$	$r_{2,1}$	$r_{2,2}$	$r_{1,3}$
$(0, 1, 0)$	4.2	2	1
...		...	
$(1, 1, 1)$	$r_{8,1}$	$r_{8,2}$	$r_{8,3}$



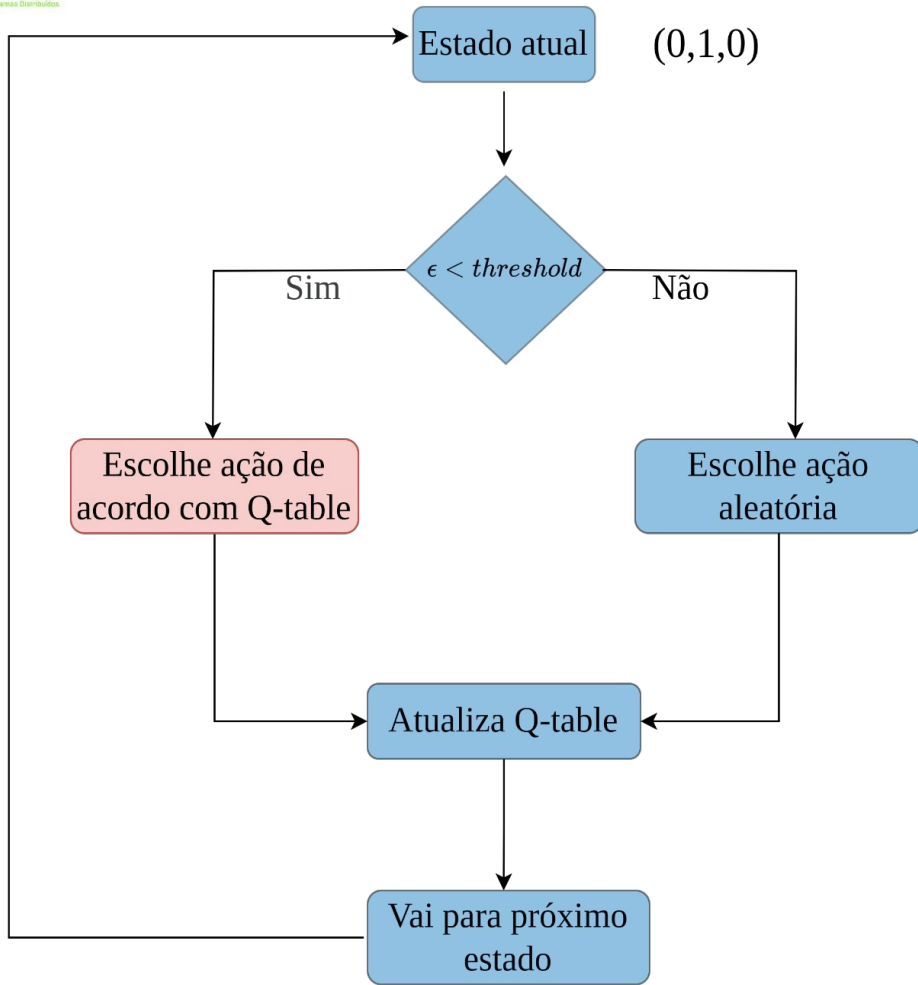


Ação selecionada!

Estados	Ações		
	A ₁	A ₂	A ₃
(0, 0, 0)	$r_{1,1}$	$r_{1,2}$	$r_{1,3}$
s_f (0, 0, 1)	$r_{2,1}$	$r_{2,2}$	$r_{1,3}$
(0, 1, 0)	3	2	1
...		...	
(1, 1, 1)	$r_{8,1}$	$r_{8,2}$	$r_{8,3}$

A tabela me informa que a ação associada a maior recompensa é A1, ou seja, inserir células!

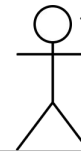




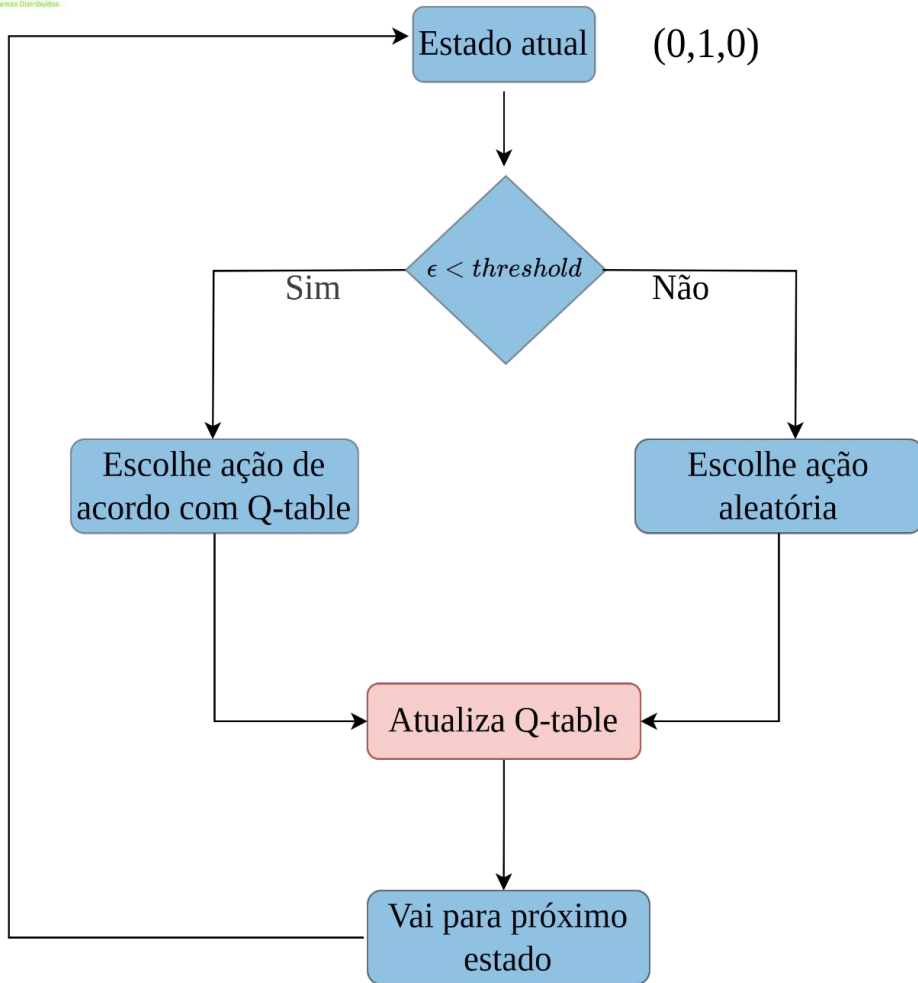
Ações

Estados	A_1	A_2	A_3
$(0, 0, 0)$	$r_{1,1}$	$r_{1,2}$	$r_{1,3}$
$s_f (0, 0, 1)$	$r_{2,1}$	$r_{2,2}$	$r_{1,3}$
$(0, 1, 0)$	3	2	1
...
$(1, 1, 1)$	$r_{8,1}$	$r_{8,2}$	$r_{8,3}$

Meu estado é $(0,1,0)$, então vou inserir $0 + 1 + 0 = 1$ célula.

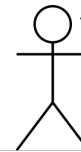


Agente

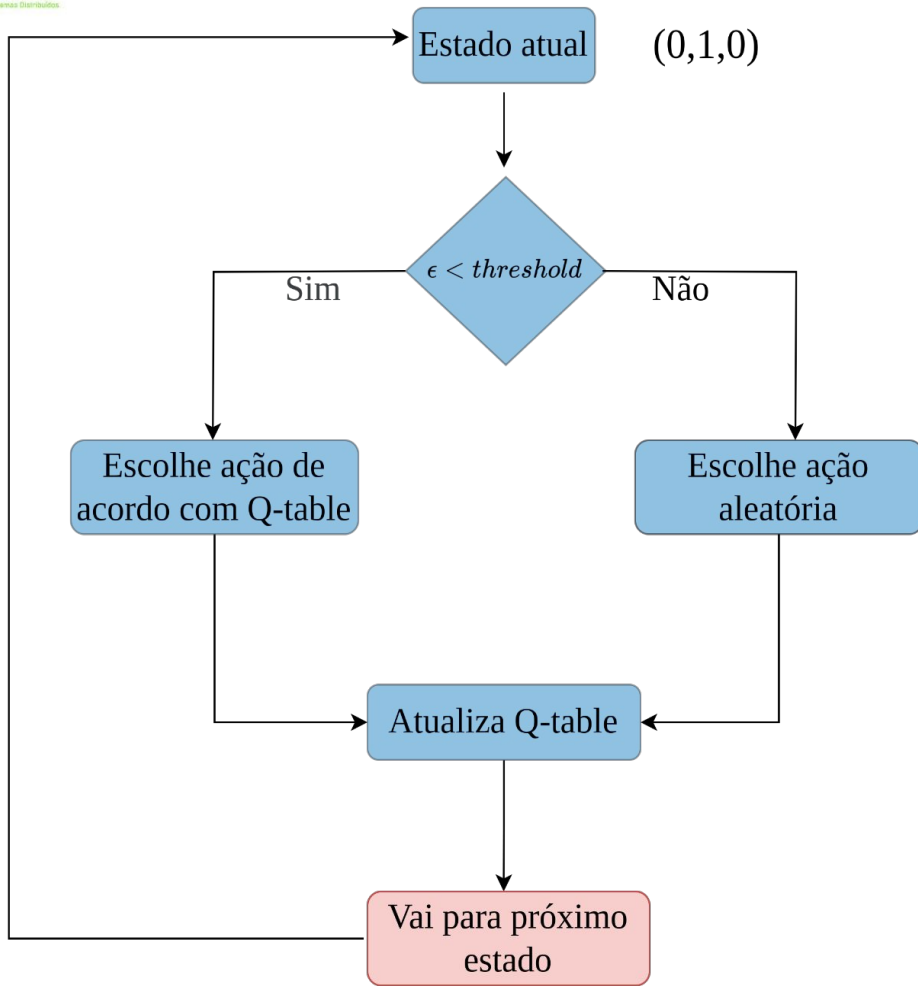


Estados	Ações		
	A ₁	A ₂	A ₃
(0, 0, 0)	$r_{1,1}$	$r_{1,2}$	$r_{1,3}$
s_f (0, 0, 1)	$r_{2,1}$	$r_{2,2}$	$r_{1,3}$
(0, 1, 0)	4.2	2	1
...
(1, 1, 1)	$r_{8,1}$	$r_{8,2}$	$r_{8,3}$

Preciso atualizar a Q-table com a equação de Bellman!



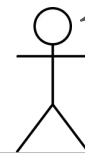
Agente



Ações

Estados	A_1	A_2	A_3
$(0, 0, 0)$	$r_{1,1}$	$r_{1,2}$	$r_{1,3}$
$s_f (0, 0, 1)$	$r_{2,1}$	$r_{2,2}$	$r_{1,3}$
$(0, 1, 0)$	4.2	2	1
...		...	
$(1, 1, 1)$	$r_{8,1}$	$r_{8,2}$	$r_{8,3}$

Meu novo estado é (0,0,0).



Agente

Experimentos

Cenários	
Aplicação	Periódica, Inundação
Topologia	Malha, Aleatória
Número de dispositivos	10, 50, 100, 150, 200

- Implementado no 6TiSCH Simulator.
- Comparado com a *Minimal Scheduling Function (MSF)*.
- Cada combinação foi executada 10 vezes.
- As métricas resultantes foram agregadas através da média, com intervalo de confiança de 99%.

Experimentos

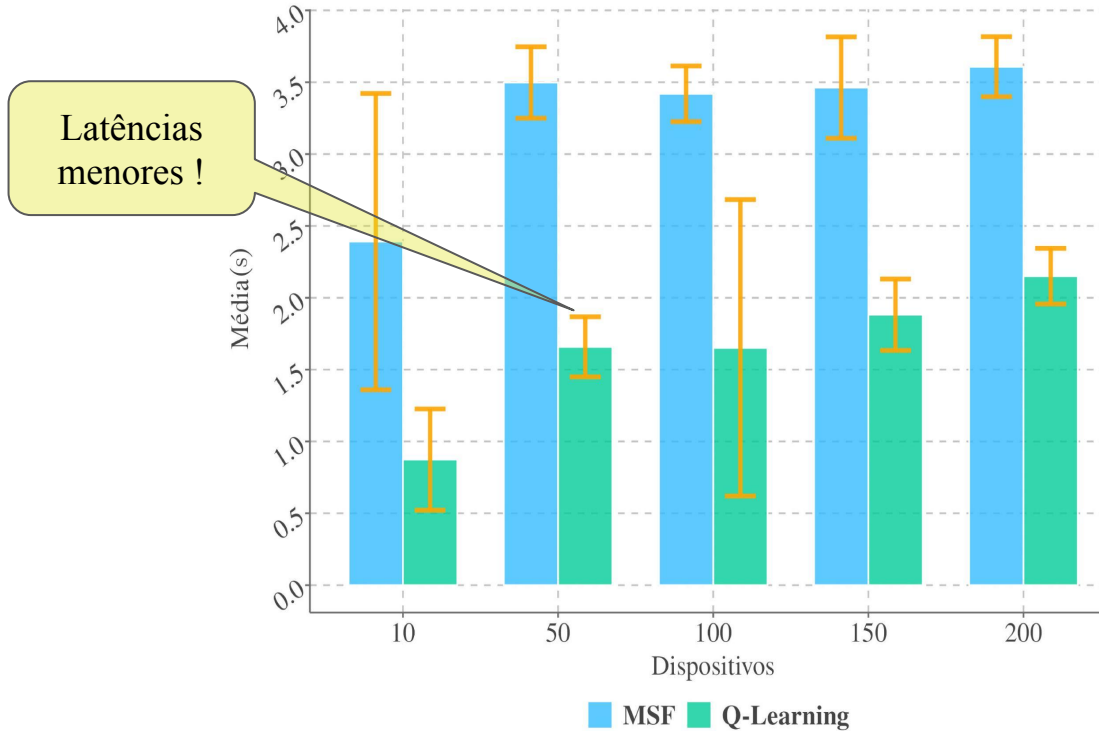
Parâmetros	Valores
Tamanho da Fila	5
Duração do Experimento	3750 Slotframes
Tamanho do Timeslot	10 ms
Tamanho do Slotframe	101 timeslots
Canais Disponíveis	1-16
α (Q-Learning)	0.7
γ (Q-Learning)	0.3

Resultados

Foram utilizadas 4 métricas para avaliar o método:

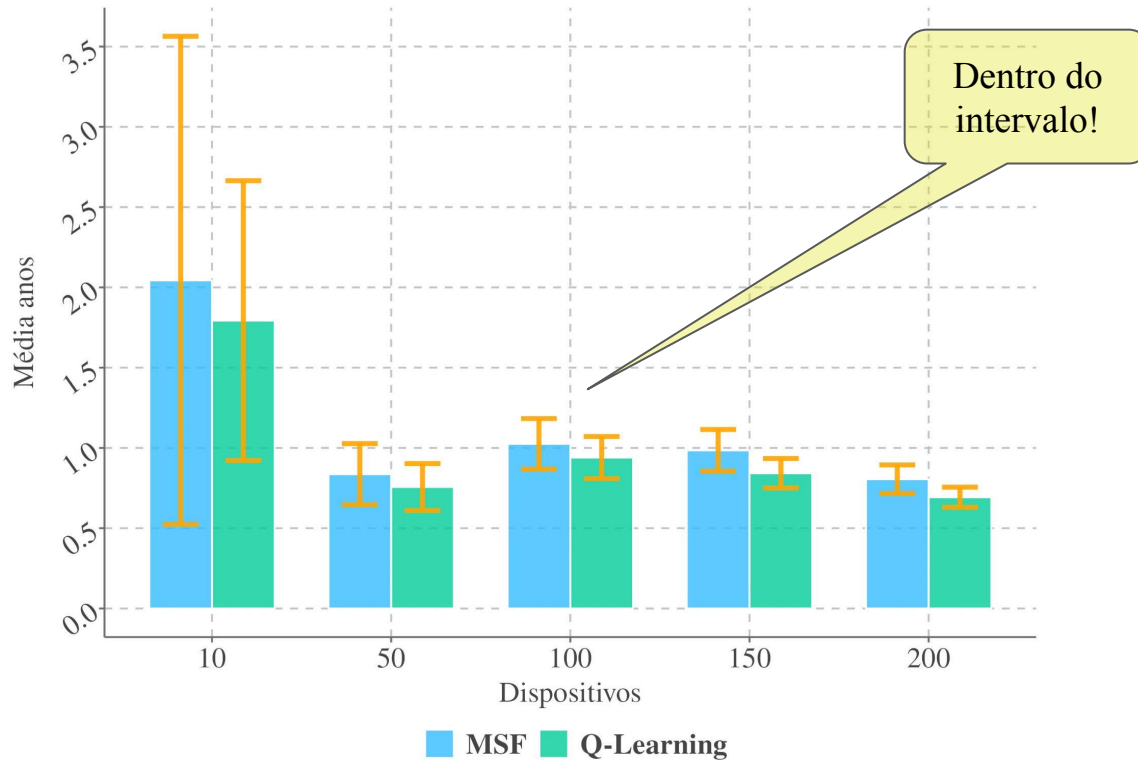
- Latência (segundos)
- Tempo de vida da rede (anos)
- Taxa de entrega dos pacotes
- Tempo de adesão à rede (segundos)

Latências



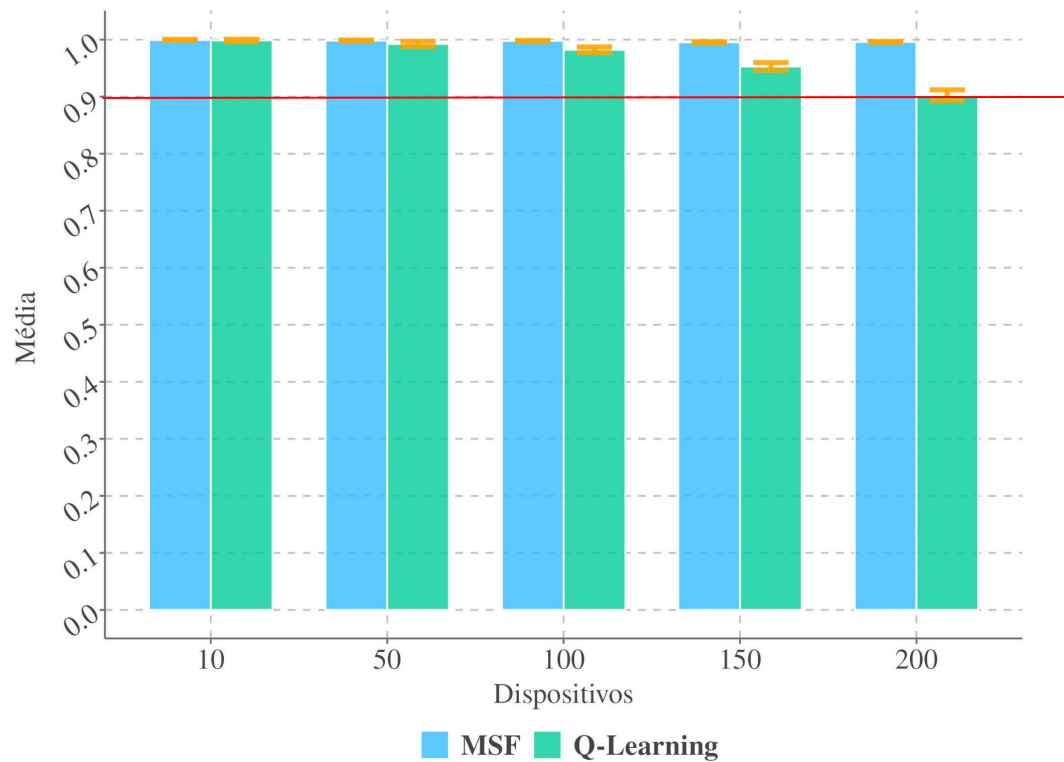
Topologia randômica com inundação.

Tempo de Vida da Rede



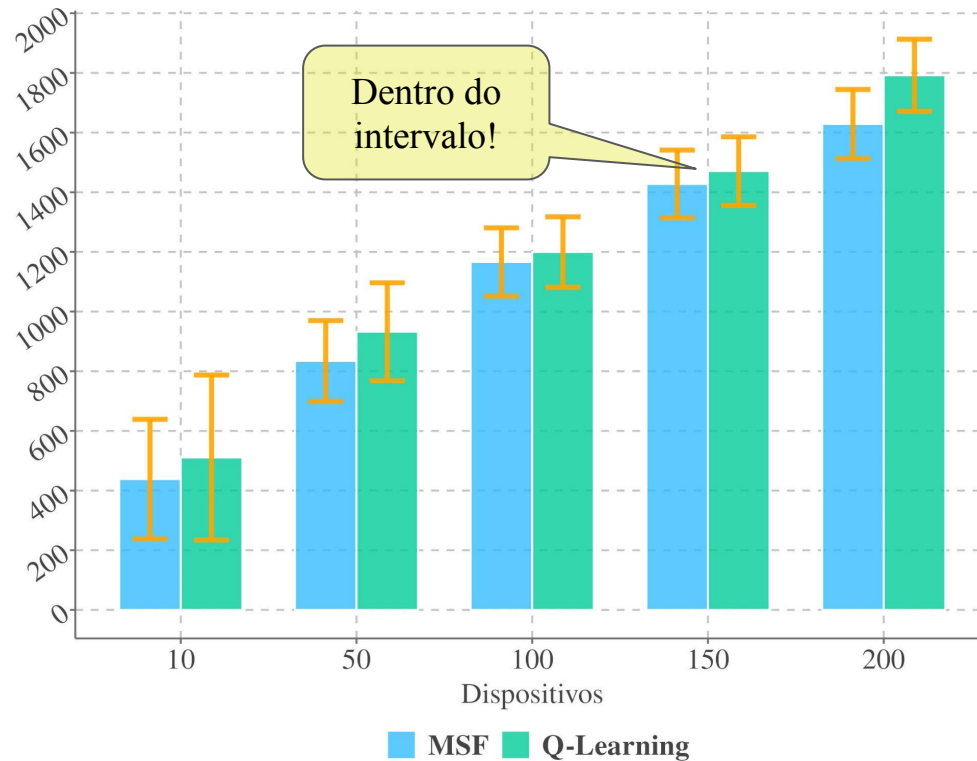
Topologia randômica com inundação.

Taxa de Entrega



Taxa de entrega - malha/periódica

Tempo de Associação à rede



Tempo de associação - malha/periódica

Discussão

- **Latências:** Houve uma redução significativa pela abordagem que usa Q-Learning;
- **Tempo de Vida:** Ambos obtiveram tempos de vida comparáveis dentro do intervalo de confiança;
- **Adesão à rede:** Ambos obtiveram tempos de vida comparáveis dentro do intervalo de confiança;
- **Taxa de entrega:** Taxa de entrega se manteve igual ou superior a 90% em ambos os métodos.

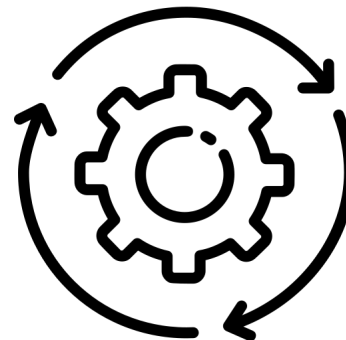
Conclusão

- **Implementamos:** Um algoritmo de escalonamento baseado em Q-Learning;
- **Metodologia de código aberto:** <https://github.com/vscardel/Q-TSCH>
- **Um comparativo qualitativo:** Dos trabalhos de escalonamento para o 6TiSCH;
- **Resultados experimentais:** Demonstrando redução da latência, enquanto mantém todas as outras métricas comparáveis com a MSF.

Trabalhos Futuros



Melhorar a performance nas outras métricas e comparar com outras metodologias



Automatizar a escolha de parâmetros.

Obrigado!

victor.cardel@ufba.br