



**UNIVERSIDADE FEDERAL DO AMAZONAS**  
**FACULDADE DE TECNOLOGIA**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

# **Mecanismo de Controle de Potência para Estimativa de Etiquetas em Redes de Identificação por Rádio Frequência**

**WALFREDO DA COSTA LUCENA FILHO**

Orientador: Prof. Dr. Celso Barbosa Carvalho

MANAUS

2015

WALFREDO DA COSTA LUCENA FILHO

MECANISMO DE CONTROLE DE POTÊNCIA PARA  
ESTIMATIVA DE ETIQUETAS EM REDES DE  
IDENTIFICAÇÃO POR RÁDIO FREQUÊNCIA

Dissertação apresentada ao Programa de Pós-Graduação em Pós-Graduação em Engenharia Elétrica da Universidade Federal do Amazonas, como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica na área de concentração Controle e Automação de Sistemas.

Aprovado em 03 de agosto de 2015.

BANCA EXAMINADORA

Prof. Dr. Celso Barbosa Carvalho, Presidente

Universidade Federal do Amazonas – UFAM

Prof. Dr. Waldir Sabino da Silva Júnior, Membro Interno

Universidade Federal do Amazonas – UFAM

Prof. Dr. Leandro Silva Galvão de Carvalho, Membro Externo

Universidade Federal do Amazonas – UFAM

## Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

L935m Lucena Filho, Walfredo da Costa  
Mecanismo de Controle de Potência para Estimativa de Etiquetas em Redes de Identificação por Rádio Frequência / Walfredo da Costa Lucena Filho. 2015  
58 f.: il. color; 31 cm.

Orientador: Celso Barbosa Carvalho  
Dissertação (Mestrado em Engenharia Elétrica) - Universidade Federal do Amazonas.

1. Redes de Computadores. 2. RFID. 3. Algoritmos Anticolisão. 4. RSSI. 5. Estimadores. I. Carvalho, Celso Barbosa II. Universidade Federal do Amazonas III. Título

*Para Alice e Gabriel.*

## **Agradecimentos**

A Deus.

À minha esposa, Ketlen, por ter estado sempre presente.

Aos meus familiares, pelo carinho, incentivo e apoio.

Ao meu orientador e amigo, Celso Carvalho, pelo incentivo, paciência e grande apoio.

Ao PPGEE/UFAM, pela oportunidade.

À FAPEAM pelo apoio financeiro.

À Fucapi pelo apoio.

Ao amigo Rodrigo Silva pelo apoio.

Ao Prof. Cláudio Gonçalves pelo apoio e compreensão.

A todos aqueles que ajudaram de alguma forma na realização deste trabalho,  
o meu mais profundo agradecimento.

## Resumo

Um sistema de identificação por rádio frequência (RFID) é composto basicamente de um leitor e etiquetas. Para que o processo de identificação das etiquetas seja bem sucedido, é necessário um algoritmo anticolisão a fim de evitar colisões entre etiquetas que respondem simultaneamente à interrogação do leitor. O algoritmo anticolisão mais usado é o DFSA (*Dynamic Framed Slotted ALOHA*) devido à sua simplicidade e baixo custo computacional. Em algoritmos probabilísticos, tal como o DFSA, o tamanho ótimo do quadro TDMA (*Time Division Multiple Access*) utilizado para leitura das etiquetas deve ser igual à quantidade de etiquetas não lidas. Uma vez que no processo de leitura, normalmente não se sabe a quantidade exata de etiquetas, o algoritmo DFSA faz uso de um estimador para obter um desempenho mais próximo do ideal. Atualmente, as aplicações têm demandado a identificação de grandes quantidades de etiquetas, o que ocasiona um aumento das colisões e, conseqüentemente, a degradação no desempenho dos algoritmos DFSA tradicionais. Este trabalho propõe um mecanismo de controle de potência para estimar a quantidade de etiquetas em redes de identificação por rádio frequência (RFID). O mecanismo baseia-se na divisão da área de interrogação em subáreas e, conseqüentemente, subgrupos de etiquetas. Tal divisão é utilizada para realizar medições de RSSI (*Received Signal Strength Indicator*) e, assim, estimar a quantidade de etiquetas por subárea. O mecanismo é simulado e avaliado utilizando um simulador próprio desenvolvido em linguagem C/C++. Neste estudo, comparam-se os resultados de quantidade de *slots* e tempo de identificação das etiquetas, com os obtidos a partir da utilização dos algoritmos DFSA ideal e algoritmo padrão  $Q$  da norma EPCglobal. A partir dos resultados da simulação, é possível perceber que o mecanismo proposto apresenta desempenho 99% do DFSA ideal em redes densas, onde há grande quantidade de etiquetas. Em relação ao algoritmo  $Q$ , percebe-se a melhoria de 6,5% no desempenho. É importante ressaltar também a redução no consumo de energia do leitor em torno de 63% em relação ao DFSA ideal.

**Palavras-chave:** Redes de Computadores, RFID, Algoritmos Anticolisão, DFSA, Controle de potência, RSSI, Estimadores.

## Abstract

An RFID system is typically composed of a reader and a set of tags. An anti-collision algorithm is necessary to avoid collision between tags that respond simultaneously to a reader. The most widely used anti-collision algorithm is DFSA (Dynamic Framed Slotted ALOHA) due to its simplicity and low computational cost. In DFSA algorithms, the optimal TDMA (Time Division Multiple Access) frame size must be equal to the number of unread tags. If the exact number of tags is unknown, the DFSA algorithm needs a tag estimator to get closer to the optimal performance. Currently, applications have required the identification of large numbers of tags, which causes an increase in collisions and hence the degradation in performance of the traditional algorithms DFSA. This work proposes a power control mechanism to estimate the number of tags for radio frequency identification networks (RFID). The mechanism divides the interrogation zone into subgroups of tags and then RSSI (Received Signal Strength Indicator) measurements estimate the number of tags in a subarea. The mechanism is simulated and evaluated using a simulator developed in C/C++ language. In this study, we compare the number of slots and identification time, with ideal DFSA algorithm and  $Q$  algorithm EPCglobal standard. Simulation results shows the proposed mechanism provides 99% performance of ideal DFSA in dense networks, where there are many tags. Regarding the  $Q$  algorithm, we can see the improvement in performance of 6.5%. It is also important to highlight the lower energy consumption of the reader comparing to ideal DFSA is 63%.

**Keywords:** Computer Networks, RFID, Anti-collision algorithms, DFSA, Power control, RSSI, estimators.

# SUMÁRIO

<b>Capítulo 1 .....</b>	<b>1</b>
<b>Introdução.....</b>	<b>1</b>
1.1 Motivação .....	1
1.2 Objetivos e metodologia .....	3
1.3 Trabalhos Relacionados .....	3
1.4 Organização do Trabalho.....	5
<b>Capítulo 2.....</b>	<b>6</b>
<b>Fundamentação Teórica.....</b>	<b>6</b>
2.1 Tecnologia RFID .....	6
2.1.1 Frequências de Operação.....	7
2.1.2 Arquitetura RFID.....	9
2.1.3 Padronização.....	10
2.2 Protocolos Anticolisão .....	11
2.2.1 Algoritmo ALOHA clássico.....	12
2.2.2 Algoritmo <i>Slotted</i> ALOHA (SA).....	12
2.2.3 Algoritmo <i>Framed Slotted</i> ALOHA (FSA).....	12
2.2.4 Algoritmo <i>Dynamic Framed Slotted</i> ALOHA (DFSA) .....	13
2.3 Estimadores.....	13
2.3.1 Schoute .....	14
2.3.2 Vogt .....	14
2.3.3 Chen.....	16
2.3.4 Eom.....	17
2.4 Algoritmo <i>Enhanced Dynamic Framed Slotted</i> ALOHA (EDFSA).....	18
2.5 Algoritmo <i>Q</i> .....	19
2.6 Técnica de controle de potência para esquemas anticolisão em RFID ..	22
2.7 Técnica de estimativa baseada na intensidade do sinal em RFID .....	25



2.8 Modelo de propagação para leitor e etiquetas .....	28
<b>Capítulo 3 .....</b>	<b>30</b>
<b>Mecanismo proposto de estimativa de etiquetas.....</b>	<b>30</b>
3.1 Divisão em <i>clusters</i> .....	30
3.2 Medições RSSI .....	33
3.2 Algoritmo Proposto.....	34
<b>Capítulo 4 .....</b>	<b>37</b>
<b>Experimentos e Resultados .....</b>	<b>37</b>
4.1 Simulador .....	37
4.2 Cenários de simulação .....	37
4.3 Resultados Obtidos .....	39
4.3.1 Quantidade de <i>slots</i> .....	42
4.3.2 <i>Throughput</i> .....	44
4.3.3 Tempo de identificação das etiquetas .....	47
4.3.4 Energia consumida pelo leitor .....	50
<b>Capítulo 5 .....</b>	<b>54</b>
<b>Conclusões e Trabalhos Futuros .....</b>	<b>54</b>
5.1 Conclusões .....	54
5.2 Trabalhos Futuros .....	55
<b>Referências Bibliográficas.....</b>	<b>56</b>

## LISTA DE ACRÔNIMOS

<i>BS</i>	<i>Binary Search</i>
<i>CDMA</i>	<i>Code Division Multiple Access</i>
<i>CW</i>	<i>Continuous Wave</i>
<i>DFSA</i>	<i>Dynamic Framed Slotted ALOHA</i>
<i>EPC</i>	<i>Electronic Product Code</i>
<i>FDMA</i>	<i>Frequency Division Multiple Access</i>
<i>FSA</i>	<i>Frame Slotted ALOHA</i>
<i>ID</i>	<i>Identificador</i>
<i>IoT</i>	<i>Internet of Things</i>
<i>ISM</i>	<i>Industrial Scientific Medical (faixa de frequência)</i>
<i>ISO</i>	<i>International Organization for Standardization</i>
<i>QT</i>	<i>Query Tree</i>
<i>RSSI</i>	<i>Received Signal Strength Indicator</i>
<i>RFID</i>	<i>Radio Frequency IDentification</i>
<i>RN16</i>	<i>16-bit Random ou pseudo-random number</i>
<i>SA</i>	<i>Slotted ALOHA</i>
<i>SDMA</i>	<i>Space Division Multiple Access</i>
<i>TDMA</i>	<i>Time Division Multiple Access</i>

## LISTA DE FIGURAS

Figura 2.1 – Técnica <i>Backscatter</i> .....	6
Figura 2.2 – Arquitetura de um sistema RFID .....	9
Figura 2.3 – Classificação dos protocolos anticolisão.....	11
Figura 2.4 – Diagrama de tempo para o EPC Gen 2. Fonte: EPC (2013). .....	20
Figura 2.5 – Algoritmo $Q$ . Fonte: EPC (2013). .....	21
Figura 2.6 – Um exemplo de uma zona de interrogação de três <i>clusters</i> . .....	24
Figura 2.7 – Codificação de bit e detecção de colisão usando codificação <i>Manchester</i> . .....	26
Figura 2.8 – Potências entre um leitor e uma etiqueta .....	28
Figura 3.1 – Zona de interrogação dividida em <i>clusters</i> . .....	31
Figura 3.2 – Distância mínima $R_{\min}$ e máxima $R_{\max}$ de um <i>cluster</i> . .....	31
Figura 3.3 – Zona de interrogação dividida em dois <i>clusters</i> . .....	34
Figura 3.4 – Algoritmo Proposto. ....	36
Figura 4.1 – Cenário de simulação. ....	38
Figura 4.2 – Quantidade de etiquetas versus Quantidade ótima de <i>clusters</i> . .....	40
Figura 4.3 – Quantidade de <i>clusters</i> versus Quantidade de <i>slots</i> para 5.000 etiquetas. ....	40
Figura 4.4 – Erros de Estimativa por <i>cluster</i> .....	41
Figura 4.5 – Quantidade de etiquetas versus Quantidade de <i>slots</i> (100 a 1.000). .....	42
Figura 4.6 – Quantidade de etiquetas versus Quantidade de <i>slots</i> (2.000 a 20.000). .....	43
Figura 4.7 – Quantidade de etiquetas versus <i>throughput</i> (%). .....	46
Figura 4.8 – Quantidade de etiquetas versus Tempo de identificação. ....	49
Figura 4.9 – Quantidade de etiquetas versus Consumo de energia do leitor. ....	52

## LISTA DE TABELAS

Tabela 2.1 – Faixa de frequências .....	7
Tabela 2.2 – Faixa de leitura por frequência .....	8
Tabela 2.3 – Tamanho do quadro ótimo para uma dada faixa de etiquetas. ....	16
Tabela 2.4 – Quantidade de etiquetas não lidas <i>versus</i> tamanho do quadro ótimo mod M. ....	19
Tabela 2.5 – Combinações das intensidades dos sinais.....	27
Tabela 3.1 – Parâmetros usados na Equação (3.2). ....	32
Tabela 3.2 – Potências de referência dos <i>clusters</i> . ....	33
Tabela 4.1 – <i>Clusters</i> que utilizam o algoritmo $Q$ . $R_f = 1m$ .....	39
Tabela 4.2 – Quantidade de <i>slots</i> para o algoritmo $Q$ e mecanismo proposto. ....	44
Tabela 4.3 – <i>Throughput</i> para algoritmo $Q$ e mecanismo proposto. ....	45
Tabela 4.4 – Tempo de Identificação para algoritmo $Q$ , mecanismo e DFSA ideal. ....	48
Tabela 4.5 – Consumo de energia do leitor para algoritmo $Q$ , mecanismo e DFSA ideal... ..	51

# Capítulo 1

## Introdução

### 1.1 Motivação

Um sistema de identificação de objetos por rádio frequência (RFID) é composto por um servidor (*middleware*), um leitor ou leitores, e um grupo de etiquetas (FINKENZELLER, 2010). Uma etiqueta passiva não possui energia própria e utiliza a energia transmitida pelo leitor para refletir o sinal com seu número de identificação.

Os sistemas RFID apresentam a vantagem de poder ler os números de identificação (ID) das etiquetas sem a necessidade de um contato próximo ou em linha de visada direta, tal como em etiquetas de códigos de barras, leitores biométricos, *smart cards*, etc. Os sistemas RFID também permitem a identificação, rastreamento e monitoramento de objetos em diversas aplicações, por exemplo, no gerenciamento de cadeia de suprimentos, manufatura, gerenciamento de estoques, Internet das Coisas (*Internet of Things*, IoT) e em qualquer aplicação onde objetos se movam de um local para o outro. Aplicações que demandem o uso de uma grande quantidade de etiquetas, por exemplo, Internet das Coisas (IoT), necessitam do menor tempo de identificação possível (GLOVER, 2006).

A comunicação entre o leitor e as etiquetas é feita por um único canal. Por isso, se duas ou mais etiquetas tentam transmitir ao mesmo tempo o leitor detecta uma colisão e, para que as etiquetas sejam lidas, é necessário utilizar um algoritmo anticolisão. Os algoritmos anticolisão são divididos em dois grupos (KLAIR *et al.*, 2010): os probabilísticos, baseados no protocolo ALOHA e os determinísticos, baseados em árvore. Os algoritmos baseados em ALOHA são os mais utilizados nos sistemas RFID comerciais, pois apresentam um custo computacional mais baixo do que os baseados em árvore. A menor utilização dos algoritmos baseados em árvore decorre da baixa capacidade de processamento e pouca memória das etiquetas, o que dificulta a utilização de algoritmos mais complexos.

Dentre os algoritmos baseados em ALOHA, o mais utilizado atualmente é o DFSA (*Dynamic Framed Slotted ALOHA*). Estes algoritmos utilizam um quadro TDMA dividido em *slots* de tempo durante os quais as etiquetas respondem com seus números de

identificação (ID). Os algoritmos necessitam de uma estimativa precisa da quantidade de etiquetas a serem lidas para que possam enviar o tamanho do quadro ótimo.

Existe uma grande variedade de métodos, chamados de estimadores (SCHOUTE, 1983) (VOGT, 2002) (CHEN, 2012), que estimam a quantidade de etiquetas em uma determinada área de leitura, conhecida como zona de interrogação. Esta estimativa permite que se obtenha o melhor *throughput*, ou seja, a máxima eficiência na leitura (BUENO-DELGADO *et al.*, 2009). Nos algoritmos DFSA, se o tamanho do quadro for maior ou menor do que a quantidade de etiquetas não lidas, a quantidade de *slots* necessários para a leitura será maior do que o valor do DFSA ideal.

Em ambientes de alta densidade de etiquetas, os algoritmos anticolisão e os mecanismos tradicionais apresentam uma eficiência menor ao interrogar as etiquetas, uma vez que há um aumento no número de colisões. Novas abordagens estão sendo propostas para reduzir as limitações destes mecanismos (SU *et al.*, 2010), como por exemplo, dividir a área de interrogação em subáreas, denominadas *clusters*, utilizando o controle de potência do leitor para reduzir as colisões (ALSALIH *et al.*, 2013). Pode-se também utilizar as medições RSSI (*Received Signal Strength Indicator*) para estimar a quantidade de etiquetas (ALOTAIBI *et al.*, 2010). Entretanto, esses trabalhos não abordam a utilização de estimadores simples e com baixo custo computacional. Assim, a relevância deste estudo seria o de propor uma solução que poderia aproveitar as vantagens da divisão da zona de interrogação em conjunto com a medição das potências retornadas pelas etiquetas de cada subgrupo. Esta nova abordagem permite uma estimativa precisa e de baixo custo computacional da quantidade de etiquetas na zona de interrogação. Isto proporcionaria um melhor desempenho do sistema em decorrência da redução da quantidade de *slots* total, e conseqüentemente do menor tempo de identificação e do menor consumo de potência do leitor. A redução do consumo de potência do leitor se deve ao aumento gradual da potência transmitida para a leitura das etiquetas nos vários *clusters*. Ou seja, quando o leitor interroga as etiquetas mais próximas, a potência necessária será menor do que a potência para interrogar as etiquetas mais distantes. Diferentemente da abordagem sem divisão da zona de interrogação, que consiste em utilizar a máxima potência durante todo o processo de interrogação e identificação das etiquetas.

Por meio de simulações, este trabalho analisa os erros de estimativa, em função da quantidade de divisões da zona de interrogação (*clusters*), e determina os parâmetros mais adequados para a redução do tempo de identificação de acordo com a quantidade de etiquetas.

A partir dos resultados da simulação, é possível perceber que o mecanismo proposto apresenta desempenho de 99% do DFSA ideal em redes densas, onde há grande quantidade de etiquetas. Em relação ao algoritmo  $Q$ , utilizado pelo padrão EPCglobal, percebe-se a melhoria de 6,5% no desempenho. É importante ressaltar também a redução no consumo de energia do leitor em torno de 63% em relação ao DFSA ideal.

## 1.2 Objetivos e metodologia

O objetivo deste trabalho é reduzir o tempo de identificação de etiquetas em uma zona de interrogação. Para isto, é proposto um mecanismo de estimativa da quantidade de etiquetas que utiliza o controle de potência do leitor e a medição dos sinais refletidos pelas etiquetas. Os objetivos específicos são os seguintes:

- Construir um simulador próprio em linguagem C/C++ para testar, validar e avaliar os algoritmos e mecanismos anticollisão pesquisados.
- Realizar experimentos para os mecanismos e os algoritmos anticollisão mais utilizados e avaliar seus desempenhos em cenários próximos aos reais.
- Avaliar o mecanismo proposto por meio da comparação com o algoritmo DFSA ideal e outros algoritmos anticollisão DFSA.

## 1.3 Trabalhos Relacionados

Foram selecionados alguns trabalhos relacionados com o tema desta dissertação e que ofereceram contribuições relevantes para nossas pesquisas. Há muitos estudos na literatura para aumentar o desempenho dos protocolos anticollisão baseados em ALOHA (EOM e LEE, 2010), (LEE, 2005), (BUENO-DELGADO *et al.*, 2009), (JIA *et al.*, 2012), GUILAN e GUOCHAO, 2010). Nossos estudos se inspiraram basicamente em três pesquisas: (ALSALIH *et al.*, 2013), (ALOTAIBI *et al.*, 2010) e (DAKHAL e SHIN, 2013), detalhados a seguir:

O trabalho de Alsalih *et al.* (2013) apresentou o método de controle de potência para esquemas anticolisão em sistemas RFID. A técnica parte da observação de que os algoritmos anticolisão existentes fazem a leitura simultânea de todas as etiquetas em uma zona de interrogação, o que proporciona um aumento das colisões quando há uma grande quantidade de etiquetas. Esta nova abordagem propõe que a zona de interrogação seja dividida em *clusters*, ou seja, agrupamentos de etiquetas que são lidas separadamente de acordo com a distância ao leitor. O estudo conclui que a probabilidade de colisão é reduzida, pois a quantidade de etiquetas em um *cluster* é menor do que em toda a zona de interrogação. Os experimentos elaborados pelos autores demonstram que a divisão em *clusters* obtém um melhor desempenho em algoritmos determinísticos do que em algoritmos probabilísticos baseados no ALOHA. O esquema proposto, entretanto, não considera a maior demanda por processamento dos algoritmos determinísticos e não propõe nenhuma alteração nos algoritmos baseados em ALOHA para aproveitar as vantagens da divisão da zona de interrogação.

O trabalho de Alotaibi *et al.* (2010) propõe um método para estimar a quantidade de etiquetas baseado na intensidade do sinal recebido RSSI (*Received Signal Strength Indicator*) durante as colisões. É proposta uma equação, válida somente na codificação *Manchester*, para estimar a quantidade de etiquetas a partir das informações obtidas com as colisões. O método também apresenta a vantagem de utilizar medidores de intensidade de sinal (potência) simples para a implementação do algoritmo. Os múltiplos sinais transmitidos pelas etiquetas chegam simultaneamente ao leitor. O total da intensidade do sinal recebido em um *slot* é a soma dos sinais individuais transmitidos pelas etiquetas. No entanto este trabalho está limitado somente à codificação *Manchester*, e apesar da simplicidade e rapidez nas medições o algoritmo apresenta um custo computacional alto.

Dakhal e Shin (2013) apresentaram um mecanismo de agrupamento de etiquetas por setor e baseado em potência (SPG – *Sector and Power based Grouping*). Este algoritmo separa as etiquetas dentro da zona de interrogação em pequenos grupos (setores) que são interrogados sequencialmente. O agrupamento de grandes conjuntos de etiquetas em grupos menores reduz a taxa de colisão e permite alocar um tamanho de quadro que aumente a eficiência do sistema. Entretanto, este mecanismo utiliza um sistema complexo de controle das antenas.



## 1.4 Organização do Trabalho

Esta dissertação está organizada em cinco capítulos. No Capítulo 2, são apresentados os princípios básicos de um sistema RFID, algoritmos anticollisão, estimadores, o padrão EPCglobal e o modelo de propagação entre o leitor e a etiqueta.

O Capítulo 3 detalha o mecanismo proposto de controle de potência para estimar a quantidade de etiquetas em uma determinada área, por meio da medição dos sinais retornados pelas etiquetas ao leitor.

O Capítulo 4 descreve detalhes do cenário de simulação e apresenta os resultados destes experimentos avaliando a quantidade de *slots*, o tempo de interrogação das etiquetas e o consumo de energia do leitor.

Finalmente, o Capítulo 5 traz as conclusões do trabalho, discute as contribuições do trabalho desenvolvido e apresenta sugestões de trabalhos futuros.

## Capítulo 2

### Fundamentação Teórica

Este capítulo discute os conceitos básicos relacionados à tecnologia RFID que foram utilizados durante a pesquisa.

#### 2.1 Tecnologia RFID

As origens dos sistemas de Identificação por Rádio Frequência (RFID) podem ser encontradas na Segunda Guerra Mundial quando os alemães aprimoraram o radar para diferenciar seus aviões das aeronaves inimigas (DOBKIN, 2012). Os pilotos realizavam uma manobra de rolagem (alternar as asas em torno do eixo longitudinal do avião) quando retornavam à base, fazendo com que o sinal refletido tivesse sua amplitude alterada.

A consequente modulação do sinal refletido permitia que os operadores de radar alemães identificassem seus aviões. Segundo Dobkin (2012), este exemplo é considerado a primeira utilização de um *link* de rádio utilizando a técnica *backscatter* passiva (Figura 2.1) para identificação. A técnica *backscatter* passiva refere-se à ausência de um rádio transmissor no objeto a ser identificado. O sinal usado para a comunicação é o sinal propagado pela estação transmissora e refletido (*scattered*) de volta pelo objeto.

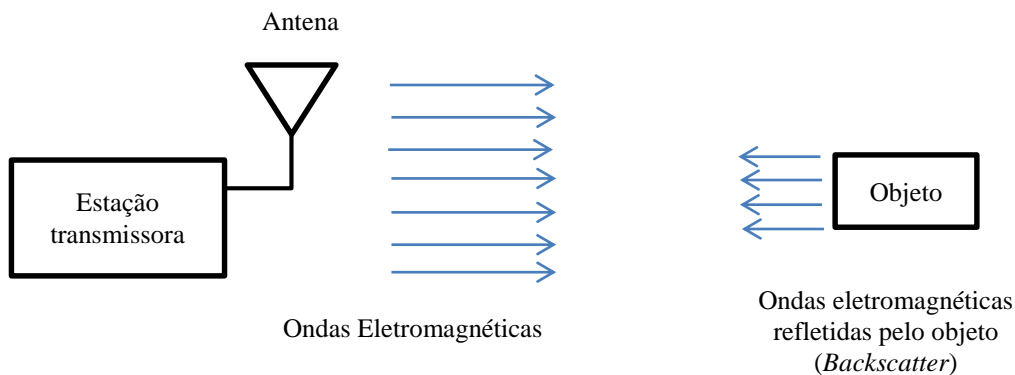


Figura 2.1 - Técnica *Backscatter*

### 2.1.1 Frequências de Operação

Os sistemas RFID utilizam o seguinte espectro de frequência (FINKENZELLER, 2010):

- LF (*low frequency* – baixa frequência);
- HF (*high frequency* – alta frequência);
- UHF (*ultra-high frequency* – ultra-alta frequência);
- Micro-ondas.

Como os sistemas de RFID transmitem ondas eletromagnéticas, eles são regulamentados como dispositivos de rádio. Estes sistemas não devem interferir em outros serviços tais como os de emergência ou transmissões de sinais de televisão.

Tabela 2.1 - Faixa de frequências

Nome	Faixa de Frequência	Frequências ISM
LF	30–300 kHz	< 135 kHz
HF	3–30 MHz	6,78 MHz; 13,56 MHz; 27,125 MHz; 40,68 MHz
UHF	300 MHz–3 GHz	433,92 MHz; 869 MHz; 915 MHz
Micro-ondas	> 3 GHz	2,45 GHz; 5,8 GHz; 24,125 GHz

Sistemas RFID operam em uma faixa não licenciada do espectro (Tabela 2.1), muitas vezes referida como ISM (*Industrial Scientific Medical - frequency range*). Entretanto, as frequências exatas que constituem a ISM podem variar dependendo da regulamentação de cada país.

Frequências diferentes possuem propriedades diferentes. Sinais de frequências mais baixas propagam melhor através da água, enquanto que as frequências mais elevadas podem transportar mais informações. Sinais de maior frequência também são mais fáceis de ler à distância. A Tabela 2.2 mostra as faixas de leitura para as diferentes frequências e como elas têm sido utilizadas em diferentes aplicações (GLOVER, 2006).

Tabela 2.2 – Faixa de leitura por frequência

<b>Frequência</b>	<b>Distância máxima</b>	<b>Aplicações</b>
LF	50 cm	Identificação de animais e leitura próxima de itens com alto conteúdo de água
HF	3 m	Controle de acesso
UHF	10 m	Caixas e paletes
Micro-ondas	> 10 m	Identificação de veículos

Em resumo, as faixas de leitura por frequência seguem as especificações abaixo (GLOVER, 2006):

- **LF RFID**
  - Usada para rastreamento de animais.
  - 125–134 kHz.
  - Acoplamento Indutivo, até 1,5m.
- **HF RFID**
  - Passaporte, cartões de crédito, livros e sistemas de pagamento sem contato.
  - 13,56 MHz.
  - Acoplamento Indutivo, até 1m.
- **UHF RFID**
  - Usada em paletes de armazenamento e em algumas carteiras de habilitação.
  - Banda de 902-928 MHz (EUA, Brasil).
  - Vários metros (10m).
- **Micro-ondas**
  - Usada em identificação de veículos, pedágio eletrônico.
  - 2,45 GHz.
  - Acima de 10m.

## 2.1.2 Arquitetura RFID

Um sistema RFID é composto por um conjunto de etiquetas, um leitor ou vários leitores e um *middleware* (Figura 2.2). Um leitor de RFID identifica objetos por meio da leitura dos dados contidos nas etiquetas (FINKENZELLER, 2010).

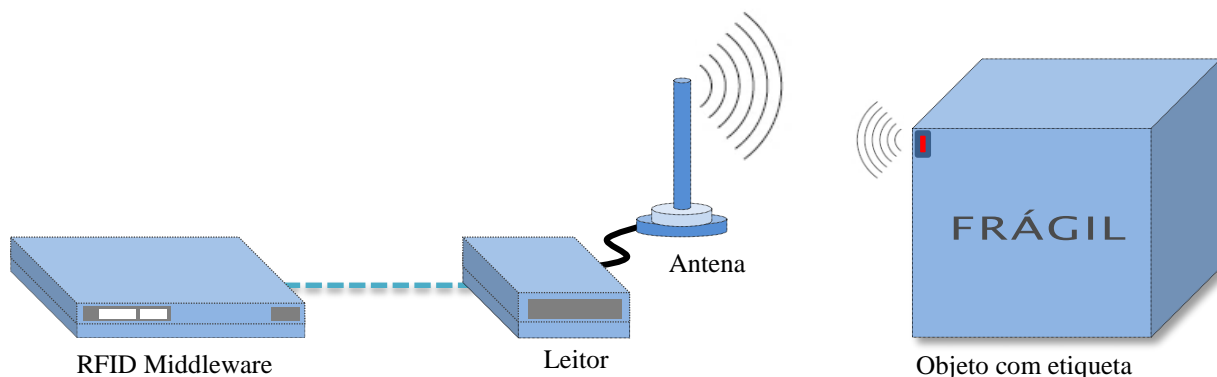


Figura 2.2 - Arquitetura de um sistema RFID

Os leitores de RFID têm como objetivo identificar etiquetas o mais rápido possível. Uma grande dificuldade no processo de leitura das etiquetas é que somente uma etiqueta pode ser lida de cada vez. Desta forma, em cenários com várias etiquetas, por exemplo, etiquetas em um palete, a leitura torna-se problemática quando várias delas respondem simultaneamente. Isto ocorre porque os respectivos sinais colidem e se corrompem entre si. Para solucionar este problema, os leitores de RFID executam um protocolo anticisão para arbitrar as repostas das etiquetas para que as colisões sejam mínimas ou não existam, e, conseqüentemente, obter a identificação das etiquetas em um menor tempo.

Um importante critério para diferenciar os sistemas RFID é quanto à fonte de alimentação das etiquetas. As etiquetas podem ser classificadas de três maneiras (DOBKIN, 2012):

- Passivas;
- Semi-passivas;
- Ativas.

As etiquetas passivas não possuem fonte de energia própria para alimentar o circuito da etiqueta e seu transmissor. Elas dependem do sinal recebido do leitor para alimentação e transmissão do sinal refletido pela etiqueta (*backscatter*).

As etiquetas semi-passivas possuem uma bateria para alimentar o circuito da etiqueta, mas continuam a utilizar a potência refletida pela etiqueta (*backscatter*) para realizar a comunicação, ou seja, necessitam da energia do leitor para o retorno do sinal.

As etiquetas ativas possuem uma bateria que alimenta tanto o circuito da etiqueta como o seu transmissor, formando assim uma configuração convencional de um sistema de comunicação bidirecional.

### **2.1.3 Padronização**

As duas principais organizações de padronização de sistemas RFID (DOBKIN, 2012) são a ISO (*International Organization for Standardization*) e a EPCglobal.

A EPCglobal é responsável por criar e controlar o número de identificação único (UID) em cada etiqueta produzida no mundo, que é conhecido como EPC (*Electronic Product Code*) (EPC, 2013). Esse código possibilita identificar o fabricante, o tipo de produto, o número de série e outras informações necessárias para rastreamento em uma cadeia produtiva.

As ISO 15693 e ISO 14443 padronizam as etiquetas HF. O padrão EPCglobal Gen 2 foi adotado como padrão global ISO 18000-6C. O ISO 18000-7 é utilizado como padrão internacional para etiquetas ativas operando em 433 MHz (DOBKIN, 2012).

A norma ISO 18000-6 é um padrão internacional que especifica a maneira que os leitores e as etiquetas se comunicam na faixa de UHF. Atualmente, existem três versões do padrão: 18000-6A, 18000-6B e 18000-6C (DOBKIN, 2012). Destas versões, a 18000-6C é a mais utilizada e a que será estudada neste trabalho.

## 2.2 Protocolos Anticolisão

Quando várias etiquetas transmitem seus IDs para um leitor ao mesmo tempo, os sinais das etiquetas colidem e impedem o leitor de reconhecer estas etiquetas. As colisões aumentam o atraso no reconhecimento das etiquetas uma vez que várias retransmissões são necessárias para o envio de seus IDs. Por isso, a utilização de protocolos anticolisão é necessária para minimizar este problema (KLAIR *et al.*, 2010).

Em um sistema RFID existe outro tipo de colisão chamada de colisão entre leitores e ocorre quando dois ou mais leitores interrogam uma etiqueta simultaneamente. Os sinais dos leitores colidem e a etiqueta não pode decodificar o sinal.

Os protocolos anticolisão podem ser agrupados em duas categorias (FINKENZELLER, 2010): protocolos baseados em ALOHA e protocolos baseados em árvore, conforme a Figura 2.3. Em geral, protocolos baseados em árvore são determinísticos, apresentam um custo computacional elevado, requerem bastante memória e necessitam de um *hardware* mais complexo. Por outro lado, os protocolos baseados em ALOHA são mais simples de implementar em *hardware*, têm menor complexidade, apresentam um menor número de comandos enviados pelo leitor e são capazes de se adaptar dinamicamente às variações na quantidade de etiquetas.

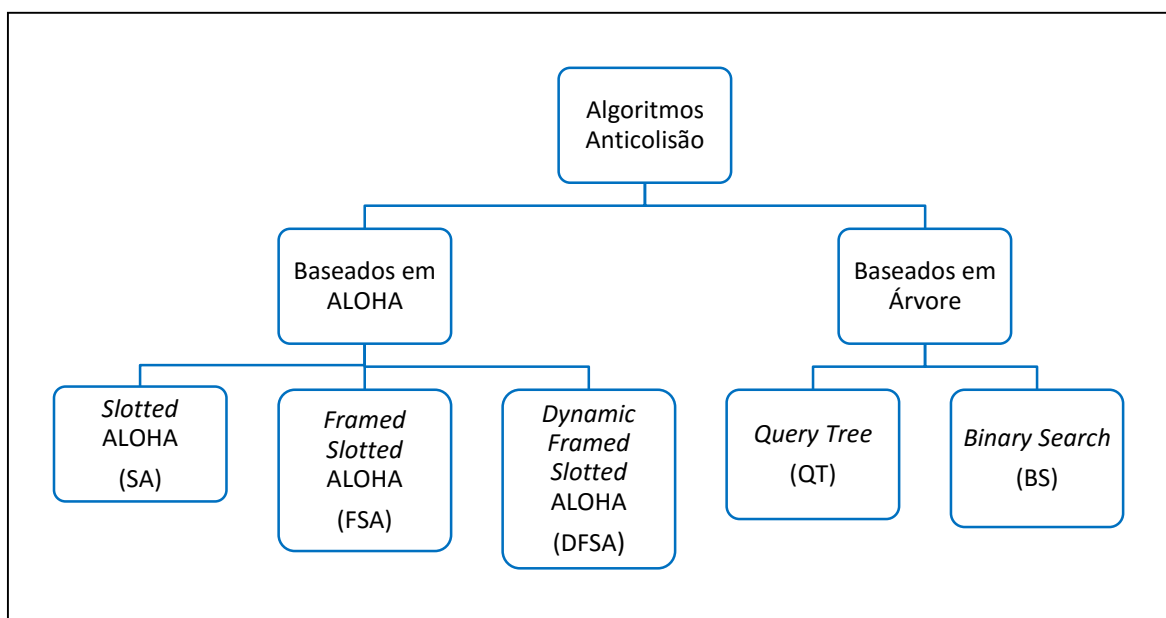


Figura 2.3 - Classificação dos protocolos anticolisão.

### 2.2.1 Algoritmo ALOHA clássico

A etiqueta, ao ser energizada, se configura para responder em um tempo aleatório após receber um comando do leitor (BOLIC, 2010). Se o leitor recebe a resposta da etiqueta corretamente, ele transmite de volta um comando de reconhecimento ACK. Caso contrário, há uma colisão e o leitor transmite um comando de negativa de reconhecimento NACK, que faz a etiqueta retransmitir seu ID após um tempo aleatório. Sistemas RFID baseados no protocolo ALOHA clássico apresentam o problema das colisões parciais que limitam o seu *throughput* a 18% (FINKENZELLER, 2010).

Com o objetivo de melhorar o desempenho do ALOHA clássico, foram propostas melhorias para reduzir as colisões e aumentar a taxa de identificação das etiquetas, por exemplo, o ALOHA clássico com *Muting*. Nesta variação, o leitor envia um comando de *mute* após a identificação da etiqueta, fazendo com que ela não responda aos próximos comandos de leitura (BOLIC, 2010).

### 2.2.2 Algoritmo *Slotted* ALOHA (SA)

A taxa de leitura do protocolo ALOHA clássico é limitada pela existência das colisões parciais. Uma maneira de resolver este problema é utilizar o protocolo *Slotted* ALOHA (BOLIC, 2010), que tem um *throughput* de 36,8%, duas vezes maior que no ALOHA clássico.

Neste protocolo, ao invés da resposta da etiqueta ser em um tempo aleatório, as etiquetas respondem em *slots* pré-definidos. Assim, o leitor e as etiquetas estão sincronizados fazendo com que as etiquetas transmitam somente no início de cada *slot*. Se ocorrer uma colisão, as etiquetas esperam por um número aleatório de *slots* antes de retransmitirem. O ganho no desempenho do *Slotted* ALOHA decorre do fato de que as colisões ocorrem somente no início de cada *slot*, diferentemente do ALOHA clássico onde as colisões podem acontecer em qualquer instante.

### 2.2.3 Algoritmo *Framed Slotted* ALOHA (FSA)

A desvantagem dos protocolos ALOHA clássico e *Slotted* ALOHA reside no fato das etiquetas poderem responder mais de uma vez em um ciclo de leitura. Para solucionar este



problema, foi criado o protocolo *Framed Slotted ALOHA* (FSA) (BOLIC, 2010) que utiliza quadros para garantir que a resposta da etiqueta aconteça somente uma vez em cada ciclo de leitura. Com isso, há uma redução na quantidade de colisões. Além disso, o método permite que o leitor estime a quantidade de etiquetas e a quantidade de *slots* ou o tamanho do quadro necessário para reduzir as colisões e *slots* vazios.

#### **2.2.4 Algoritmo *Dynamic Framed Slotted ALOHA* (DFSA)**

O protocolo FSA possui os quadros de tamanho fixo. Isto resulta em uma desvantagem, pois o leitor irá detectar uma grande quantidade de *slots* vazios quando a quantidade de etiquetas para leitura for pequena. Por outro lado, o leitor irá detectar muitos *slots* com colisão quando a quantidade de etiquetas lidas aumenta muito em comparação ao tamanho do quadro, o que diminui o desempenho do protocolo. Para reduzir este problema, foi desenvolvido o protocolo DFSA (*Dynamic Framed Slotted ALOHA*) (SCHOUTE, 1983) para permitir que o tamanho do quadro seja ajustado dinamicamente de acordo com a quantidade de etiquetas.

O protocolo DFSA é o mais utilizado dos algoritmos baseados em ALOHA (CHEN, 2012). O máximo *throughput* do algoritmo DFSA ocorre quando o tamanho do quadro é igual à quantidade de etiquetas restantes (SCHOUTE, 1983). Deste modo, o leitor ajusta dinamicamente o próximo tamanho do quadro de acordo com a quantidade de etiquetas não lidas.

Existe uma grande variedade de métodos para estimar a quantidade de etiquetas não lidas em uma zona de interrogação (KLAIR *et al.*, 2010). Esta estimativa permite que seja determinado o tamanho do quadro ótimo para algoritmos anticisão baseados em DFSA.

### **2.3 Estimadores**

O tamanho do quadro em um algoritmo DFSA deve ser ajustado por uma função de estimativa. Uma função de estimativa de etiquetas baseia-se na quantidade de *slots* vazios, de *slots* com colisão e de *slots* com sucesso do quadro que está sendo lido (quadro atual). A partir destes valores, a função estima o novo tamanho do quadro utilizado para a leitura das

etiquetas. Nesta seção, são apresentados alguns estimadores tradicionais utilizados em algoritmos DFSA.

### 2.3.1 Schoute

O método para estimar a quantidade de etiquetas proposto por Schoute (1983) baseia-se na quantidade de colisões no quadro atual e considera uma distribuição de *Poisson*. A quantidade de etiquetas estimadas é dada pela seguinte equação:

$$\text{quantidade de etiquetas estimadas} = c_1 + 2,39 c_k \quad (2.1)$$

onde, na Equação (2.1), as variáveis são descritas da seguinte maneira:

2,39 é o valor esperado calculado da quantidade de *slots* com colisão.

$c_1$  = quantidade de *slots* com sucesso.

$c_k$  = quantidade de *slots* com colisão.

O método de Schoute é bastante simples de implementar e apresenta baixo custo computacional. Entretanto, só deve ser utilizado quando a quantidade de etiquetas é pequena.

### 2.3.2 Vogt

O método proposto por Vogt (2002) apresenta duas funções de estimativa chamadas de Vogt-I e Vogt-II.

A função de estimativa Vogt-I, também conhecida como *lower bound*, baseia-se no princípio de que um *slot* com sucesso, ocorrido no quadro atual, possui somente uma etiqueta e que um *slot* com colisão possui duas ou mais etiquetas. Por isso, é possível estimar um limite mínimo para a quantidade de etiquetas dado pela equação:

$$\text{quantidade de etiquetas estimadas} = c_1 + 2c_k \quad (2.2)$$

onde, na Equação (2.2), as variáveis são descritas da seguinte maneira:

$c_0$  = quantidade de *slots* vazios.

$c_1$  = quantidade de *slots* com sucesso.

$c_k$  = quantidade de *slots* com colisão (múltiplas colisões).

A função de estimativa Vogt-II propõe uma abordagem diferente. Além dos valores (medidos) da quantidade de *slots* vazios ( $c_0$ ), da quantidade de *slots* com sucesso ( $c_1$ ) e da quantidade de *slots* com colisão ( $c_k$ ), no quadro atual, a função utiliza os valores esperados (calculados) destes *slots* para estimar o tamanho ótimo do próximo quadro.

A função Vogt-II é baseada na desigualdade de *Chebyshev* e tem como objetivo minimizar o erro ( $\varepsilon_{vd}$ ). Este erro é definido como a diferença entre o valor medido no quadro atual e valor esperado calculado. Utiliza-se a seguinte equação:

$$\varepsilon_{vd}(N, c_0, c_1, c_k) = \min_n \left| \begin{pmatrix} a_0^{N,n} \\ a_1^{N,n} \\ a_k^{N,n} \end{pmatrix} - \begin{pmatrix} c_0 \\ c_1 \\ c_k \end{pmatrix} \right| \quad (2.3)$$

onde, na Equação (2.3), as variáveis são descritas da seguinte maneira:

$\varepsilon_{vd}$  = erro entre a leitura atual e o resultado teórico calculado.

$N$  = tamanho do quadro.

$n$  = quantidade de etiquetas.

$\langle c_0, c_1, c_k \rangle$  = vetor com as medições da quantidade de *slots* no quadro atual.

$\langle a_0^{N,n}, a_1^{N,n}, a_k^{N,n} \rangle$  = vetor com os valores esperados para a quantidade de *slots* vazios, quantidade de *slots* com sucesso e quantidade de *slots* com colisões.

Resumindo, o estimador Vogt-II calcula o valor de  $n$  (quantidade de etiquetas estimadas) que minimiza a função erro  $\varepsilon_{vd}$ .

Os elementos do vetor  $\langle a_0^{N,n}, a_1^{N,n}, a_k^{N,n} \rangle$  correspondem aos valores esperados da quantidade de *slots*. Para o cálculo de  $a_r^{N,n}$ , dado um tamanho de quadro  $N$ , e  $n$  a quantidade de etiquetas, o valor esperado dos *slots* ocupados com  $r$  etiquetas é dado pela equação:

$$a_r^{N,n} = N \binom{n}{r} \left(\frac{1}{N}\right)^r \left(1 - \frac{1}{N}\right)^{n-r} \quad (2.4)$$

Algumas observações podem ser feitas sobre os estimadores Vogt-I e Vogt-II: O método Vogt-I pode estimar facilmente a quantidade de etiquetas. Entretanto, conforme a quantidade de etiquetas aumenta, o erro de estimativa aumenta proporcionalmente e consequentemente diminui o desempenho.

Diferentemente do Vogt-I, no Vogt-II quando temos poucas etiquetas o erro de estimativa aumenta. Para uma quantidade de etiquetas maior, o Vogt-II estima com mais precisão que o Vogt-I, mas a um custo computacional bem maior.

O estimador Vogt determina também qual o melhor tamanho de quadro para uma determinada faixa de etiquetas na zona de interrogação. O tamanho do quadro ótimo, de acordo com a faixa de valores, é apresentado na Tabela 2.3. Por exemplo, quando a quantidade de etiquetas está na faixa de 17 a 56, o tamanho do quadro ótimo a ser utilizado deve ser de 64 *slots*.

Tabela 2.3 - Tamanho do quadro ótimo para uma dada faixa de etiquetas.

Tamanho do quadro ótimo (N)	n (inferior)	n (superior)
16	1	9
32	10	27
64	17	56
128	51	129
256	112	$\infty$

### 2.3.3 Chen

O método proposto por Chen e Lin (2006) apresenta duas funções de estimativas: Chen-I e Chen-II. Muitos algoritmos utilizam a quantidade de *slots* com colisão  $c_k$  para estimar a quantidade de etiquetas. A função Chen-I baseia-se na quantidade de *slots* vazios

$c_0$  para estimar a quantidade de etiquetas por meio da probabilidade de achar *slots* vazios após uma rodada de leitura de um quadro. A seguinte equação é utilizada.

$$P(E) = \frac{(-1)^E N! n!}{E! N^n} \sum_{j=E}^N (-1)^j \frac{(N-j)^n}{(j-E)! (N-j)! n!} \quad (2.5)$$

Para um dado valor de  $N$  e  $E$ , podemos achar o valor de  $n$  que maximiza a probabilidade de  $P(E)$  na Equação (2.5).

A função de estimativa Chen-II calcula o valor esperado da quantidade de *slots* vazios ( $E$ ) e do valor esperado da quantidade de *slots* com sucesso ( $S$ ) de acordo com a equação (2.4). A Equação (2.6) é utilizada para estimar a quantidade de etiquetas não lidas.

$$n = (N - E - 1) \frac{S}{E} \quad (2.6)$$

### 2.3.4 Eom

Eom *et al.* (2010) apresentam um mecanismo de estimativa da quantidade de etiquetas para algoritmos DFSA. Os autores assumem que o tamanho do quadro  $L$  atual é dado pela seguinte equação:

$$L = \beta \cdot n \quad (2.7)$$

onde, na Equação (2.7), as variáveis são descritas da seguinte maneira:

$\beta$  = fator de proporcionalidade.

$n$  = quantidade de etiquetas estimadas.

Os autores consideram o valor esperado da quantidade de etiquetas transmitindo em um *slot* com colisão como  $\gamma$ . Logo, o tamanho do próximo quadro pode ser calculado de acordo com a equação:

$$\text{tamanho do próximo quadro} = n - c_1 = \gamma \cdot c_k \quad (2.8)$$

Apresentam também uma maneira de achar  $\gamma$  e  $\beta$  de modo iterativo de acordo com as Equações (2.7) e (2.8), uma vez que achar a solução de modo analítico é bastante complexa.

$$\beta_k = \frac{L}{\beta_{k-1} \cdot c_k + c_1} \quad (2.9)$$

$$\gamma_k = \frac{1 - e^{-\frac{1}{\beta_k}}}{\beta_k \left( 1 - \left( 1 + \frac{1}{\beta_k} \right) e^{-\frac{1}{\beta_k}} \right)} \quad (2.10)$$

logo, o tamanho estimado do próximo quadro é dado pela equação:

$$\text{tamanho do próximo quadro} = \gamma_{k^*} \cdot c_k \quad (2.11)$$

onde, na Equação (2.11),  $\gamma_{k^*}$  é o valor limite encontrado quando a diferença  $|\gamma_{k^*-1} - \gamma_{k^*}|$  é a menor possível.

## 2.4 Algoritmo *Enhanced Dynamic Framed Slotted ALOHA* (EDFSA)

Um problema bastante relevante em todos os protocolos FSA é o crescimento exponencial do tamanho do quadro com o aumento da quantidade de etiquetas. Os primeiros algoritmos DFSA desenvolvidos tinham uma limitação no tamanho do quadro (256 ou 512). Se a quantidade de etiquetas aumenta, ocorre um aumento das colisões. Para superar esta limitação, Lee *et al.* (2005), propuseram o algoritmo EDFSA, cujo objetivo principal é determinar o tamanho do quadro que irá maximizar a eficiência do sistema. Eficiência do sistema é definida pela seguinte Equação (2.12):

$$\text{Eficiência do Sistema} = \frac{\text{quant. esperada de slots com sucesso}}{\text{tamanho do quadro atual}} = \frac{a_1^{N,n}}{N} \quad (2.12)$$

No método EDFSA, o tamanho do quadro cresce linearmente em função da quantidade de etiquetas. Inicialmente, o algoritmo EDFSA estima a quantidade de etiquetas

não lidas por meio da função de estimativa Vogt-II e, dependendo do resultado, divide as etiquetas em  $M$  grupos chamados de Módulos conforme a Tabela 2.4. Somente um grupo de cada vez pode responder ao leitor para evitar um aumento na quantidade de colisões. Na Tabela 2.4, observa-se que quando a quantidade de etiquetas estimadas é menor que 354, o valor de  $M$  é um. Acima deste valor faz-se a divisão das etiquetas em  $M$  grupos.

Tabela 2.4 – Quantidade de etiquetas não lidas *versus* tamanho do quadro ótimo e módulo  $M$ .

Quantidade de etiquetas não lidas ( $n$ )	Tamanho do quadro ( $N$ )	Módulo ( $M$ )
:	:	:
1417 – 2831	256	8
708 – 1416	256	4
355 – 707	256	2
177 – 354	256	1
82 – 176	128	1
41 – 81	64	1
20 – 40	32	1
12-19	16	1
6-11	8	1

## 2.5 Algoritmo $Q$

O padrão EPCglobal Class-1 Gen2 (EPC, 2013) desenvolveu um protocolo anticisão DFSA denominado algoritmo  $Q$ . No esquema FSA (*Framed Slotted ALOHA*), cada quadro tem uma quantidade fixa de *slots* e as etiquetas respondem em um *slot* escolhido aleatoriamente (CHEN, 2012). O processo de interrogação das etiquetas, de acordo com a norma EPCglobal, inicia-se com o leitor transmitindo um sinal para energizar as etiquetas na zona de interrogação. Este sinal, cuja amplitude, frequência e fase são constantes, é chamado de *Continuous Wave* (CW). Um quadro individual é conhecido como uma rodada de inventário (*Inventory Round*). A sequência de comandos de interrogação, *Query*, *Ack*, *QueryRep*, etc. formam a rodada de inventário para uma etiqueta conforme a Figura 2.4.

Um comando *Query* é transmitido com alguns campos tais como os parâmetros da camada física e o parâmetro  $Q$  que determina a quantidade de *slots* em uma rodada de inventário. Quando uma etiqueta recebe um comando *Query*, ela gera um número aleatório na faixa de 0 a  $2^q - 1$  e o valor é armazenado no contador de *slot* da etiqueta.

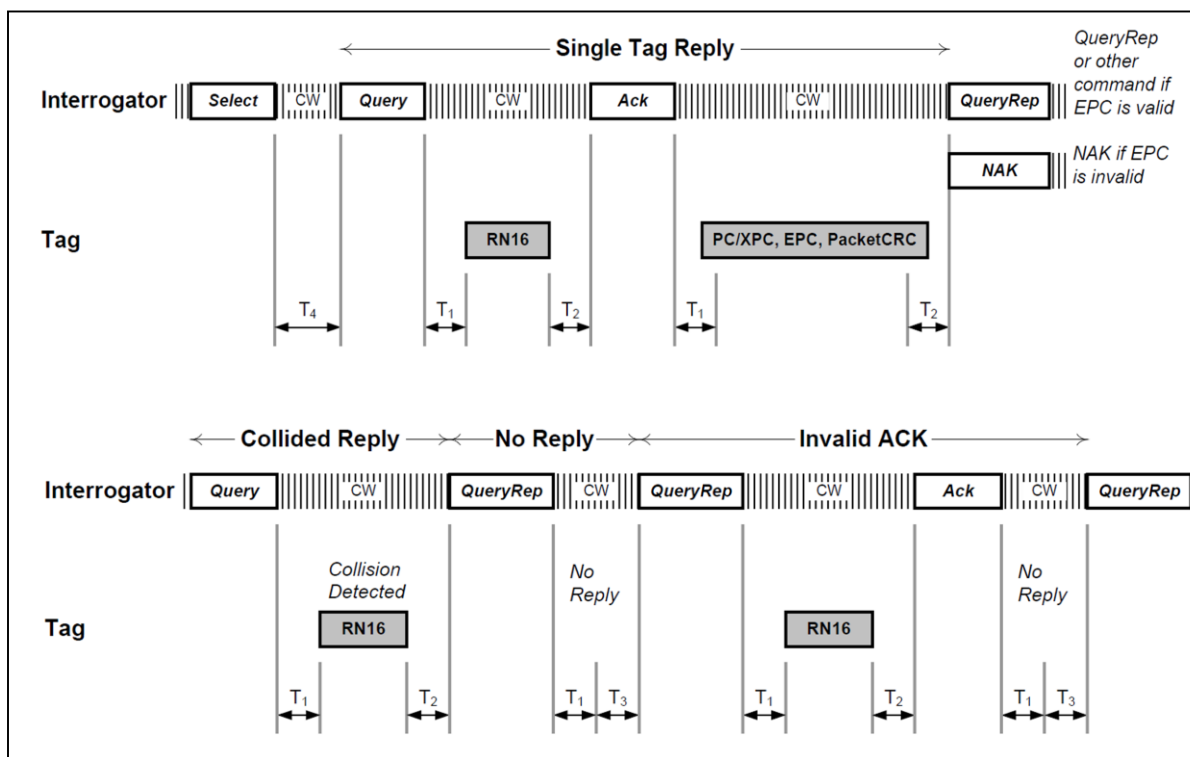


Figura 2.4 – Diagrama de tempo para o EPC Gen 2. Fonte: EPC (2013).

Quando o contador de *slot* da etiqueta for igual a 0, a etiqueta transmite imediatamente um número aleatório de 16 bits (RN16). O leitor ao receber o RN16 retorna este mesmo número em um comando *ACK*. Se a etiqueta receber de volta o mesmo RN16 significa que não houve colisão e a etiqueta pode enviar ao leitor seu EPC.

Após a transmissão do EPC pela etiqueta, um comando *QueryRep* é enviado para indicar o fim do *slot*. Depois de receber este comando, as etiquetas restantes decrementam seus contadores de *slot* e aquelas cujos contadores de *slot* forem 0 respondem com o RN16. O envio do comando *QueryRep* se repete de acordo com a quantidade de *slots* indicado pelo parâmetro  $Q$ . A parte inferior da Figura 2.4 mostra o diagrama de tempo para três etiquetas, duas das quais geraram um valor aleatório inicial igual a 0 resultando em uma colisão no primeiro *slot*. A terceira etiqueta gerou um valor aleatório inicial igual a 2 fazendo com que ela responda no segundo *QueryRep*. Como não existem etiquetas com o contador de *slot* igual a 0, o *slot* 1 está vazio.



Para evitar *slots* vazios e *slots* com colisão, o leitor altera o parâmetro  $Q$  a cada rodada de inventário para ajustar o tamanho do quadro de acordo com as etiquetas restantes. Este algoritmo é denominado algoritmo  $Q$  e está mostrado na Figura 2.5. O algoritmo  $Q$  (LEE, 2007) avalia as repostas das etiquetas, no quadro atual, e determina o próximo tamanho de quadro. O algoritmo  $Q$  classifica as repostas das etiquetas como *slot* vazio, *slot* com colisão e *slot* com sucesso, e ajusta o valor do parâmetro  $Q$  para o próximo ciclo de *Query*. O leitor mantém uma variável  $Q_{fp}$ , um número real, cujo valor é adicionado por uma constante  $C$  caso o *slot* tenha uma colisão ou o valor é subtraído por esta constante  $C$  se o *slot* estiver vazio. Em seguida, o valor de  $Q$  recebe o valor arredondado de  $Q_{fp}$  e o leitor envia um outro comando *Query* com o novo valor de  $Q$ . O valor de  $Q$  se ajusta de acordo com as colisões, se houver muitas colisões o valor de  $Q$  aumenta o tamanho do quadro, se houver muitos *slots* vazios o valor de  $Q$  diminui reduzindo o tamanho do quadro.

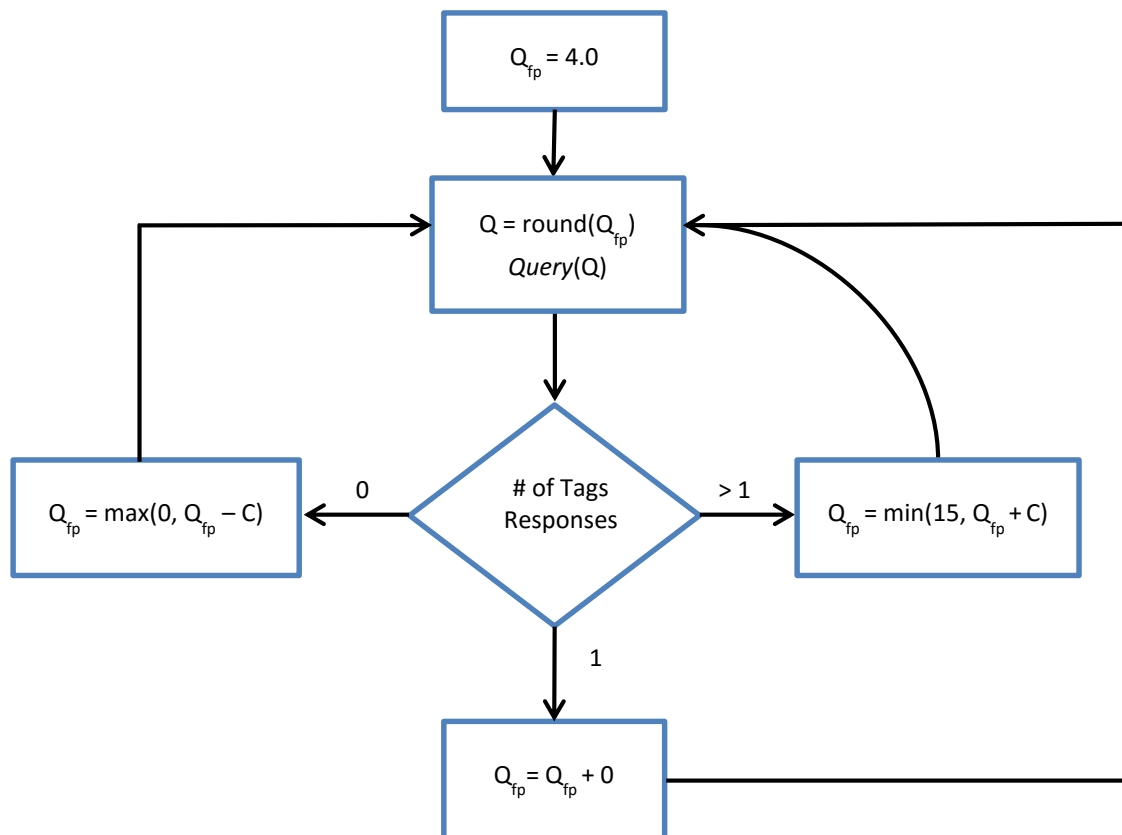


Figura 2.5 - Algoritmo  $Q$ . Fonte: EPC (2013).

## 2.6 Técnica de controle de potência para esquemas anticolisão em RFID

Os trabalhos de Alsalih *et al.* (2013) e Ali *et al.* (2007) apresentam o método de controle de potência para esquemas anticolisão em sistemas RFID. A técnica parte da observação de que os esquemas anticolisão existentes fazem a leitura simultânea de todas as etiquetas em uma zona de interrogação, o que proporciona um aumento das colisões quando há uma grande quantidade de etiquetas. Esta nova abordagem propõe que a zona de interrogação seja dividida em *clusters*, ou seja, agrupamentos de etiquetas que são lidas separadamente de acordo com a distância ao leitor. Os autores concluem que a probabilidade de colisão é reduzida, pois a quantidade de etiquetas em um *cluster* é menor do que em toda a zona de interrogação. No entanto, a quantidade de *clusters* deve ser selecionada adequadamente para que não haja muitas divisões e conseqüentemente muitos *clusters* vazios, pois estes tendem a aumentar a quantidade de ciclos de leitura. A abordagem também calcula a quantidade ideal de *clusters* para permitir a redução das colisões juntamente com a redução da quantidade de *clusters* vazios.

Na pesquisa, são considerados dois casos de otimização cujo objetivo é minimizar o tempo de identificação das etiquetas. No primeiro caso, foi elaborado um algoritmo para encontrar o melhor esquema de organização dos *clusters* considerando-se que a faixa de potências de transmissão do leitor pode ser ajustada continuamente com alta precisão. No segundo caso, considera-se o cenário onde o leitor tem um conjunto discreto de potências de transmissão. Em cada caso, foi feita uma análise matemática e elaborado um algoritmo para encontrar a quantidade ótima de *clusters*. Observa-se também que a abordagem por meio da divisão em *clusters* pode ser utilizada com qualquer esquema anticolisão (probabilísticos ou determinísticos) para melhorar o desempenho em aplicações que tenham alta densidade de etiquetas. O esquema *power-based distance clustering* (PDC) é definido como sendo o primeiro a utilizar tal abordagem. O leitor ajusta sua potência de transmissão para que as etiquetas na zona de interrogação sejam divididas em *clusters* de acordo com suas distâncias ao leitor.

A potência transmitida pelo leitor é gradativamente aumentada para que ele possa dividir em *clusters* de acordo com a distância do leitor às etiquetas. As etiquetas no padrão EPC Gen2 Class 1 possuem um *flag* que indica se elas podem ou não responder ao leitor. Os autores chamam esta condição de modo *sleep*. Cada etiqueta interrogada e identificada em

um *cluster* é colocada no modo *sleep* a fim de evitar que ela seja lida novamente ao se interrogar as etiquetas do próximo *cluster*. Uma etiqueta no modo *sleep* não responderá enquanto não for resetada pelo leitor.

Segundo os autores, o PDC é um esquema de *dividir para conquistar* (ALSALIH *et al.*, 2013), onde as etiquetas são divididas de acordo com distância ao leitor. Conseqüentemente, as etiquetas são lidas um *cluster* de cada vez, possibilitando acelerar o processo de interrogação e identificação das etiquetas. O particionamento da zona de interrogação é conseguido por meio do controle do nível de potência do leitor. A densidade de potência refletida é calculada de acordo com a equação:

$$S = \frac{\lambda^2 \cdot P_{reader} \cdot G_{reader} \cdot G_{tag}}{(4\pi)^2 R^4} \quad (2.13)$$

A área de abrangência do leitor pode ser calculada usando a seguinte equação:

$$R = \frac{\lambda}{4\pi} \sqrt[4]{\frac{k \cdot P_{reader} \cdot G_{reader}^2 \cdot G_{tag}^2}{P_{back}}} \quad (2.14)$$

onde,  $S$  é a densidade de potência refletida,  $k$  é o comprimento de onda do sinal transmitido pelo leitor,  $P_{reader}$  é a potência transmitida pela antena do leitor,  $R$  é a distância entre o leitor e a etiqueta,  $G_{reader}$  é o ganho da antena do leitor,  $G_{tag}$  é o ganho da etiqueta e  $P_{back}$  é a potência recebida pelo leitor refletida pela etiqueta. Observa-se da Equação (2.13) que a densidade potência refletida de volta para a antena do leitor é proporcional à raiz quarta da potência transmitida pelo leitor. Observa-se da Equação (2.14) que a distância de leitura entre uma etiqueta e o leitor pode ser alterada mudando-se a potência fornecida pela antena do leitor enquanto se mantém  $P_{back}$  constante. Deste modo, a faixa de interrogação do leitor pode ser reduzida diminuindo-se  $P_{reader}$ .

A Figura 2.6 mostra um exemplo da divisão da zona de interrogação em três *clusters* chamados  $D_1$ ,  $D_2$  e  $D_3$ . Quando o leitor envia uma consulta a um determinado *cluster*, somente as etiquetas deste *cluster* devem responder. Por exemplo, suponha que o leitor interrogue o *cluster*  $D_2$  após o término da leitura das etiquetas do *cluster*  $D_1$ . Somente as

etiquetas do *cluster*  $D_2$ , identificadas como  $t_2$  na Figura 2.6, irão responder ao leitor. Após todas as etiquetas  $t_2$  terem sido lidas, elas são colocadas no modo *sleep*, evitando assim que elas respondam às próximas interrogações. Em seguida, a potência do leitor é aumentada para atingir o *cluster*  $D_3$  e uma nova interrogação é enviada às etiquetas. Então, somente as etiquetas  $t_3$  do *cluster*  $D_3$  irão responder ao leitor.

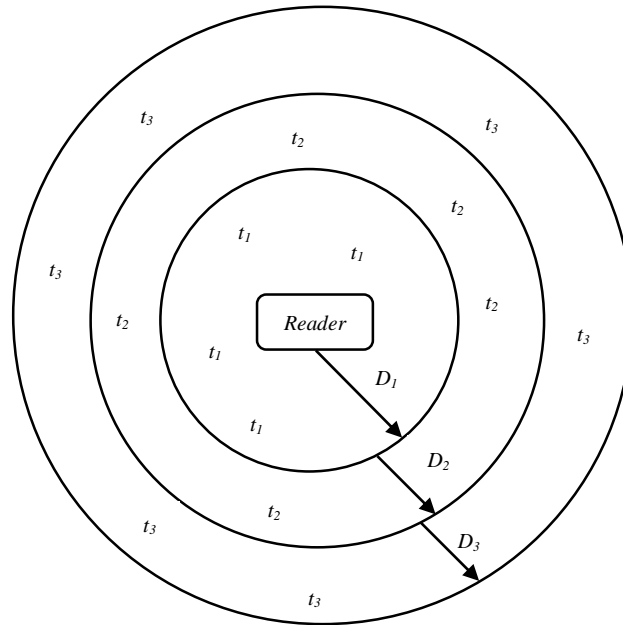


Figura 2.6 – Um exemplo de uma zona de interrogação de três *clusters*. Fonte: Alsalih *et al.* (2013)

Os experimentos elaborados por Alsalih *et al.* (2013) demonstram que a divisão em *clusters* obtém melhor desempenho em algoritmos determinísticos do que em algoritmos probabilísticos baseados no ALOHA. Segundo Alsalih *et al.* (2013), o esquema PDC com o algoritmo  $Q$  apresentou uma melhoria de 10% no desempenho em comparação ao algoritmo  $Q$  sem o mecanismo.

O trabalho de Alsalih *et al.* (2013) não apresentou nenhuma proposta de melhoria do desempenho do mecanismo PDC em conjunto com algoritmos baseados em DFSA. Devido à importância dos algoritmos DFSA, percebeu-se a necessidade de novas abordagens com estes tipos de algoritmos.

## 2.7 Técnica de estimativa baseada na intensidade do sinal em RFID

Alotaibi *et al.* (2010) apresentam um novo método para estimar a quantidade de etiquetas baseado na intensidade do sinal recebido RSS (*Received Signal Strength*) durante as colisões. Ele propõe uma equação para estimar a quantidade de etiquetas a partir das informações obtidas com as colisões. O método também apresenta a vantagem de utilizar medidores de intensidade de sinal (potência) simples para a implementação do algoritmo.

A intensidade do sinal de cada *bit* é utilizada durante um *slot* de transmissão, ou seja, se duas etiquetas transmitem ao mesmo tempo, seus sinais irão combinar somando ou subtraindo. Na implementação do algoritmo proposto, as etiquetas devem transmitir seus números de identificação (ID) usando a codificação *Manchester*, para permitir que as colisões dos *bits* sejam detectadas mais facilmente.

O modelo de simulação utilizado envolve um leitor, etiquetas e uma zona de interrogação. As etiquetas são distribuídas aleatoriamente e possuem uma distância mínima (0,5 m) e uma distância máxima (10 m) em relação ao leitor.

A magnitude da potência recebida por uma etiqueta é calculada pela equação de *Friis*:

$$P_r = P_t G_t G_r \left( \frac{\lambda}{4\pi R} \right)^2 \quad (2.15)$$

onde  $P_t$  é a potência transmitida,  $P_r$  é a magnitude da potência recebida,  $G_t$  é o ganho da antena transmissora,  $G_r$  é o ganho da antena receptora,  $\lambda$  é o comprimento de onda do sinal transmitido e  $R$  é a distância (raio) entre o leitor e a etiqueta.

Os múltiplos sinais transmitidos pelas etiquetas chegam simultaneamente ao leitor. O total da intensidade do sinal recebido em um *slot* é a soma dos sinais individuais transmitidos pelas etiquetas conforme a equação:

$$\sum_{n=0}^N P_{r,tag_n} \quad (2.16)$$

Este processo é mostrado graficamente na Figura 2.7. A detecção das colisões é relativamente fácil de ser obtida, pois utiliza-se a codificação *Manchester*.

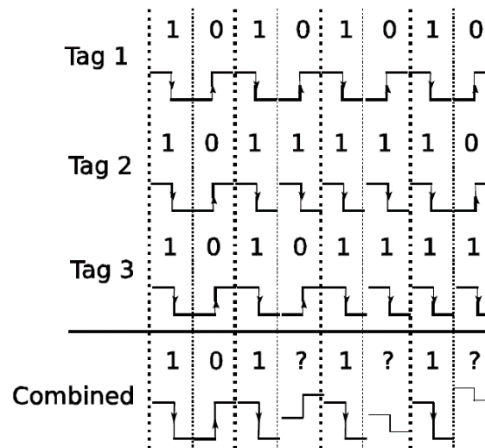


Figura 2.7 – Codificação de bit e detecção de colisão usando codificação *Manchester*.  
 Fonte: Alotaibi *et al.* (2010)

Para estimar a quantidade de etiquetas, a intensidade do sinal das colisões em dois períodos de *bits* é gravada e comparada durante um *slot*. Por exemplo, se três etiquetas (A, B, C) colidem, elas podem apresentar quatro diferentes intensidades de sinais: A+B, B+C, A+C e A+B+C. Como cada uma das colisões tem uma intensidade de sinal diferente, pode-se obter as informações necessárias para estimar a quantidade de etiquetas envolvidas na colisão. O autor propõe que o limite superior da estimativa seja  $2^n$ , onde  $n$  é a quantidade de etiquetas envolvidas na colisão. O limite inferior é dado por:

$$\operatorname{argmin}_n \left| \sum_{j=1}^{\operatorname{int}(\frac{n-1}{2})} \binom{n}{j} + E \right| \geq c_{unique} \quad (2.17)$$

$$E = \binom{n}{\frac{n}{2}} \div 2, \text{ se } n \text{ é par} \quad (2.18)$$

onde  $n$  é a quantidade estimada de etiquetas e  $c_{unique}$  é a quantidade de colisões únicas.  $E$  é o termo adicional para uma quantidade par de etiquetas. A fórmula é obtida por meio da contagem dos períodos de *bit* de colisão que possuam  $j$  etiquetas com o mesmo *bit* (devido à mesma intensidade de sinal). De acordo com as Equações (2.17) e (2.18) elabora-se a Tabela

2.5 que mostra a quantidade de etiquetas, quantidades de colisões únicas e a intensidade do sinal.

Tabela 2.5 - Combinações das intensidades dos sinais. Fonte: Alotaibi *et al.* (2010)

Quantidade de etiquetas	Combinações	Componentes	Total
1		0	
2	$\binom{2}{1} \div 2$	1	1
3	$\binom{3}{1}$	3	3
4	$\binom{4}{1} + \binom{4}{2} \div 2$	4+3	7
5	$\binom{5}{1} + \binom{5}{2}$	5+10	15
6	$\binom{6}{1} + \binom{6}{2} + \binom{6}{3} \div 2$	6+15=10	31

Os experimentos elaborados por Alotaibi *et al.* (2010) demonstram que o desempenho do algoritmo se aproxima bastante do estimador DFSA ideal, ou seja, em torno de 99%.

No entanto, esse trabalho está limitado somente à codificação *Manchester*, e apesar da simplicidade e rapidez nas medições, o algoritmo de estimativa é complexo e apresenta um custo computacional elevado.

O trabalho de Alotaibi *et al.* (2010) apresentou a técnica de medição de RSSI enfatizando a simplicidade e velocidade das medições. Esta abordagem RSSI pode ser aproveitada com outras técnicas e para reduzir suas desvantagens analisadas.

## 2.8 Modelo de propagação para leitor e etiquetas

Utiliza-se o modelo de propagação no espaço livre, de acordo com a equação de Friis (FRIIS, 1946), para calcular a potência recebida em uma antena receptora como segue:

$$P_{RX} = P_{TX} G_{TX} G_{RX} \left( \frac{\lambda}{4\pi R} \right)^2 \quad (2.19)$$

A fórmula mostra que a potência recebida  $P_{RX}$  é definida em função da potência transmitida  $P_{TX}$ , da distância entre as antenas, dos ganhos das antenas e do comprimento de onda. A Figura 2.8 apresenta as potências envolvidas na comunicação entre um leitor e uma etiqueta.

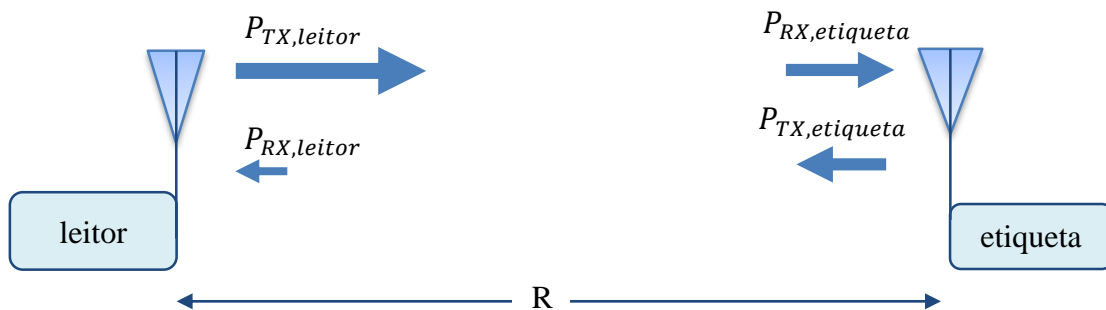


Figura 2.8 – Potências entre um leitor e uma etiqueta

Onde,

$P_{TX,leitor}$  = Potência transmitida pelo leitor.

$P_{RX,etiqueta}$  = Potência recebida na antena da etiqueta.

$P_{TX,etiqueta}$  = Potência transmitida (refletida) pela etiqueta.

$P_{RX,leitor}$  = Potência recebida (refletida) na antena do leitor.

$R$  = Distância entre as antenas do leitor e da etiqueta.

A Equação de Friis (2.19) pode ser usada para calcular (DOBKIN, 2012) o valor da potência refletida pela etiqueta no leitor ( $P_{RX,leitor}$ ). A perda por transmissão *backscatter* é denominada  $k$ .



logo,  $P_{TX,etiqueta} = P_{RX,etiqueta} \cdot k$

da equação (2.19) temos,  $P_{RX,etiqueta} = P_{TX,leitor} \cdot G_{leitor} \cdot G_{etiqueta} \left(\frac{\lambda}{4\pi R}\right)^2$

então,  $P_{TX,etiqueta} = P_{TX,leitor} \cdot G_{leitor} \cdot G_{etiqueta} \left(\frac{\lambda}{4\pi R}\right)^2 \cdot k$

da equação (2.19) temos,  $P_{RX,leitor} = P_{TX,etiqueta} \cdot G_{leitor} \cdot G_{etiqueta} \cdot \left(\frac{\lambda}{4\pi R}\right)^2$

logo, a expressão final é dada por:

$$P_{RX,leitor} = P_{TX,leitor} \cdot k \cdot G_{leitor}^2 \cdot G_{etiqueta}^2 \left(\frac{\lambda}{4\pi R}\right)^4 \quad (2.20)$$

O modelo de espaço livre é válido para etiquetas que estão no campo distante (*far-field*) da antena do leitor. O início do campo distante, também chamado de região de *Fraunhofer*, é definido (FRIIS, 1946) de acordo com a equação:

$$R_f = \frac{2D^2}{\lambda} \quad (2.21)$$

onde  $D$  é a maior dimensão da antena do leitor e  $R_f$  é a distância a partir da qual considera-se o início da região de *Fraunhofer*. Para que a Equação (2.21) seja válida  $D$  deve ser maior que o comprimento de onda  $\lambda$  ( $D > \lambda$ ).

## Capítulo 3

### Mecanismo proposto de estimativa de etiquetas

O protocolo anticolisão DFSA obtém seu desempenho ideal quando a quantidade de etiquetas estimadas é igual à quantidade de etiquetas restantes (SCHOUTE, 1983). Vários métodos para estimar a quantidade de etiquetas em uma zona de interrogação são utilizados (KLAIR *et al.*, 2010).

Neste trabalho, o mecanismo proposto de estimativa de etiquetas baseia-se na divisão da zona de interrogação em subáreas ou agrupamentos denominados *clusters* (ALSALIH *et al.*, 2013), de acordo com a Figura 3.1, e na medição das potências refletidas das etiquetas em cada *cluster*. Utilizamos um *slot* de medição (quadro com um *slot*) para medir a potência de retorno de todas as etiquetas de um *cluster*. O mecanismo proposto, nesta dissertação, tem o objetivo de diminuir o tempo de identificação de etiquetas em ambientes com alta densidade de etiquetas por meio da medição da intensidade dos sinais refletidos pelas etiquetas.

#### 3.1 Divisão em *clusters*

A largura de cada *cluster* é determinada por meio da potência transmitida pelo leitor. A divisão em *clusters* possibilita a redução da quantidade de etiquetas lidas e conseqüentemente a redução das colisões totais. A divisão em *clusters* é importante no mecanismo, pois possibilita a redução do erro de estimativa. Inicialmente, deve-se avaliar a quantidade de *clusters* utilizados, uma vez que se dividirmos em poucos *clusters* teremos um aumento no erro de estimativa e se a quantidade de *clusters* for elevada, haverá um aumento desnecessário na quantidade de *slots* de medição. Em ambos os casos, o mecanismo de estimação utiliza uma quantidade maior de *slots* para o processo de identificação, o que ocasiona a redução no desempenho do mecanismo de estimação.

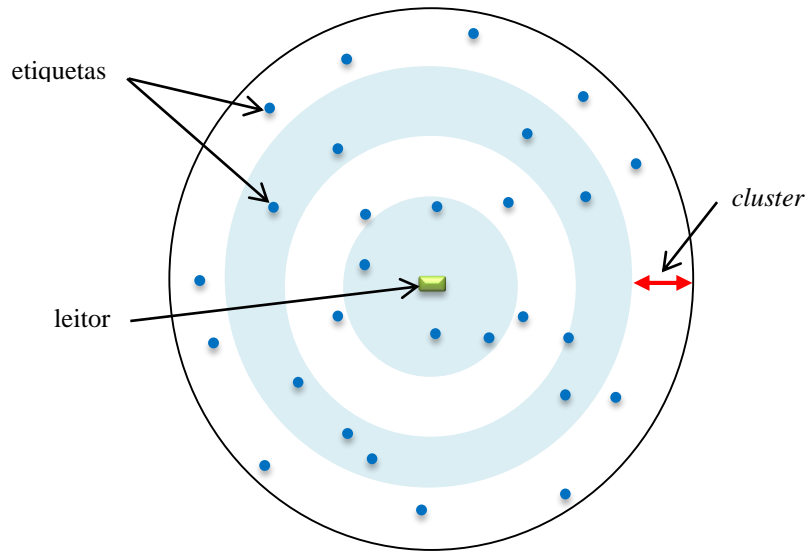


Figura 3.1– Zona de interrogação dividida em *clusters*.

A Figura 3.1 apresenta um *cluster* onde estão definidas as distância mínima do leitor  $R_{min}$  (limite inferior do *cluster*) e a distância máxima  $R_{max}$  (limite superior do *cluster*). As etiquetas que estão nesta área pertencem ao *cluster*. No mecanismo proposto, considera-se que todas as etiquetas deste *cluster* estão localizadas no centro do *cluster* cuja distância  $R_{50\%}$  é calculada pela Equação (3.1).

$$R_{50\%} = \frac{R_{min} + R_{max}}{2} \quad (3.1)$$

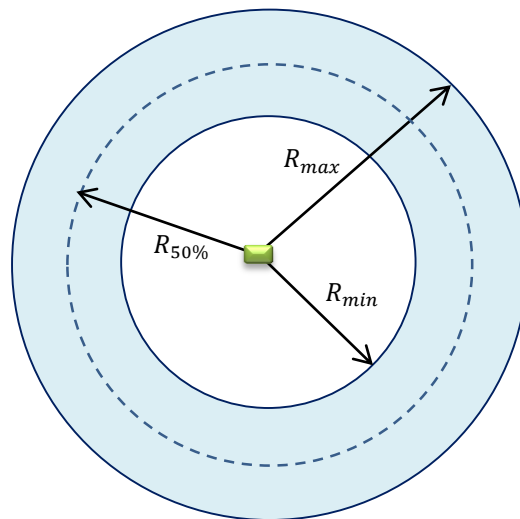


Figura 3.2 – Distância mínima  $R_{min}$  e máxima  $R_{max}$  de um *cluster*.

A partir da Equação (2.20), pode-se calcular a potência para se delimitar os *clusters*. É necessário calcular a potência de referência de cada *cluster*, denominada  $P_{cluster}$ . Esta potência define o limite superior (distância máxima) de cada *cluster*. A potência é incrementada discretamente conforme os *clusters* são lidos. Por exemplo, se dividirmos a zona de interrogação em 30 *clusters*, 30 potências  $P_{cluster}$  diferentes são necessárias para cobrir toda área a ser lida. A equação é apresentada a seguir:

$$P_{cluster} = P_{TX,leitor} = \frac{(4\pi R_{max})^4 \cdot P_{RXmin,leitor}}{\lambda^4 \cdot k \cdot G_{leitor}^2 \cdot G_{etiqueta}^2} \quad (3.2)$$

onde  $G_{leitor}$  é o ganho da antena do leitor,  $G_{etiqueta}$  é o ganho da antena da etiqueta,  $\lambda$  é o comprimento de onda em 900 MHz,  $k$  é a perda por transmissão *backscatter* e  $R_{max}$  é a distância máxima que limita cada *cluster*.  $P_{RXmin,leitor}$  é a potência mínima refletida que permite ao leitor demodular o sinal recebido, ou seja, a sensibilidade do receptor. Definimos esta potência em  $-80$  dbm. Os valores utilizados na simulação estão na Tabela 3.1,

Tabela 3.1- Parâmetros usados na Equação (3.2).

Parâmetros	Valores
$P_{RXmin,leitor}$	$10^{-11}W (-80 dBm)$
$k$	0,33
$G_{leitor}$	1
$G_{etiqueta}$	1
$\lambda$	0,33m

Como exemplo, utilizou-se uma zona de interrogação com raio de 10m dividido em 30 *clusters*, conforme (ALSALIH *et al.*, 2013), para a simulação do estimador. A Tabela 3.2 apresenta as potências de referências calculadas para delimitar cada *cluster*.

Tabela 3.2 - Potências de referência dos *clusters*.

Cluster	$R_{max}$ (m)	$P_{cluster}$ (dBm)	Cluster	$R_{max}$ (m)	$P_{cluster}$ (dBm)
0	0,33	-31,23	15	5,33	16,93
1	0,67	-19,20	16	5,67	17,98
2	1,00	-12,14	17	6,00	18,98
3	1,33	-7,15	18	6,33	19,91
4	1,67	-3,30	19	6,67	20,8
5	2,00	-0,10	20	7,00	21,65
6	2,33	2,57	21	7,33	22,46
7	2,67	4,90	22	7,67	23,23
8	3,00	6,93	23	8,00	23,97
9	3,33	8,76	24	8,33	24,68
10	3,66	10,42	25	8,67	25,36
11	4,00	11,93	26	9,00	26,02
12	4,33	13,32	27	9,33	26,65
13	4,67	14,61	28	9,66	27,26
14	5,00	15,81	29	10,00	27,85

### 3.2 Medições RSSI

O método proposto estima a quantidade de etiquetas no *cluster* por meio da medição da soma das intensidades dos sinais recebidos no *slot* de medição (quadro com um *slot*). Para a avaliação do método, o simulador calcula a potência refletida ( $P_{refletida}$ ) de cada etiqueta no *cluster* de acordo com a equação:

$$P_{refletida} = P_{RX,leitor} = \frac{P_{cluster} \cdot k \cdot G_{leitor}^2 \cdot G_{etiqueta}^2 \cdot \lambda^4}{(4\pi R)^4} \quad (3.3)$$

onde  $P_{cluster}$  é a potência de referência do *cluster* e  $R$  é a distância entre o leitor e cada etiqueta.

O mecanismo propõe que em cada *cluster*, a estimativa da quantidade de etiquetas seja usada como parâmetro para o algoritmo anticolisão DFSA. Em cada *Query*, o primeiro

quadro deve ter somente um *slot* para a medição. O leitor pode medir a potência total refletida de todas as etiquetas do *cluster* e estimar a quantidade de etiquetas não lidas. Após a estimativa, o novo tamanho de quadro do algoritmo DFSA é utilizada no *cluster*. O leitor repete este procedimento até que todas as etiquetas do *cluster* sejam lidas. Após a leitura de todos os *clusters* da zona de interrogação o procedimento é encerrado.

### 3.2 Algoritmo Proposto

Para ilustrar o funcionamento do mecanismo proposto, a Figura 3.3 mostra a divisão da zona de interrogação em dois *clusters*. O processo de leitura e estimação é descrito a seguir:

1 - O leitor envia um sinal de interrogação cuja potência transmitida ( $P_{0_{cluster}}$ ), calculada pela Equação (3.2), delimita a largura do *cluster* 0. Este sinal de interrogação informa que o tamanho do quadro é de um *slot*, chamado de *slot* de medição.

2 - Todas as etiquetas do *cluster* 0 respondem simultaneamente no *slot* de medição. Cada etiqueta transmite uma potência refletida  $P_{refletida}$ , calculada para fins de simulação de acordo com a Equação (3.3).

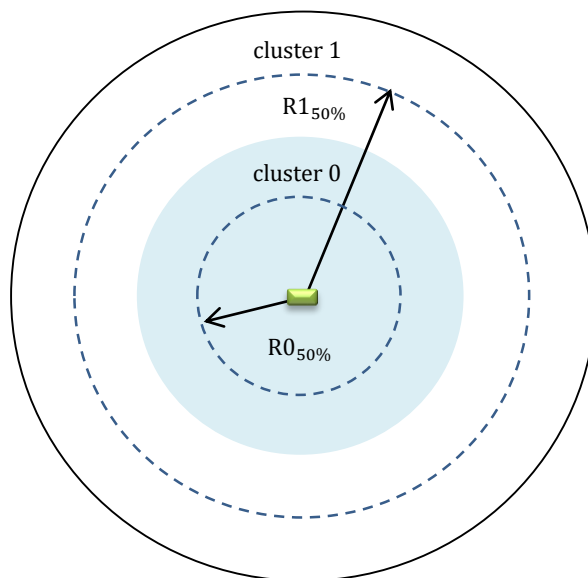


Figura 3.3 – Zona de interrogação dividida em dois *clusters*.

3 - A soma das potências refletidas pelas etiquetas ( $\sum P_{refletida}$ ) do *cluster*, que colidem no *slot* de medição, é medida pelo leitor para o cálculo da estimativa da quantidade de etiquetas no *cluster*.

4 - Para o cálculo da estimativa da quantidade de etiquetas utiliza-se a seguinte equação:

$$\text{quantidade de etiquetas estimadas} = \frac{\sum P_{refletida}}{P_{refletida,50\%}} \quad (3.4)$$

onde  $P_{refletida,50\%}$  é a potência calculada pela Equação (3.3), considerando que todas as etiquetas estão localizadas no meio do *cluster*. Na Figura 3.3, a potência refletida é calculada considerando-se a distância  $R0_{50\%}$  para o *cluster* 0 e  $R1_{50\%}$  para o *cluster* 1.

5 – Com o resultado da Equação (3.4), o leitor pode definir o tamanho do próximo quadro de acordo com a quantidade de etiquetas estimadas. As etiquetas identificadas são colocadas no modo *sleep* para que não respondam a novas consultas.

6 – Com o término da identificação de todas as etiquetas do *cluster* 0, o leitor aumenta a potência para  $P1_{cluster}$  a fim de delimitar a largura do *cluster* 1. E repete os procedimentos feitos anteriormente para o *cluster* 0.

7- Após a leitura de todos os *clusters*, o leitor envia um comando de *reset* para que todas as etiquetas da zona de interrogação saiam do modo *sleep* e passem a responder normalmente às consultas do leitor.

O fluxograma do algoritmo do mecanismo proposto é mostrado na Figura 3.4.

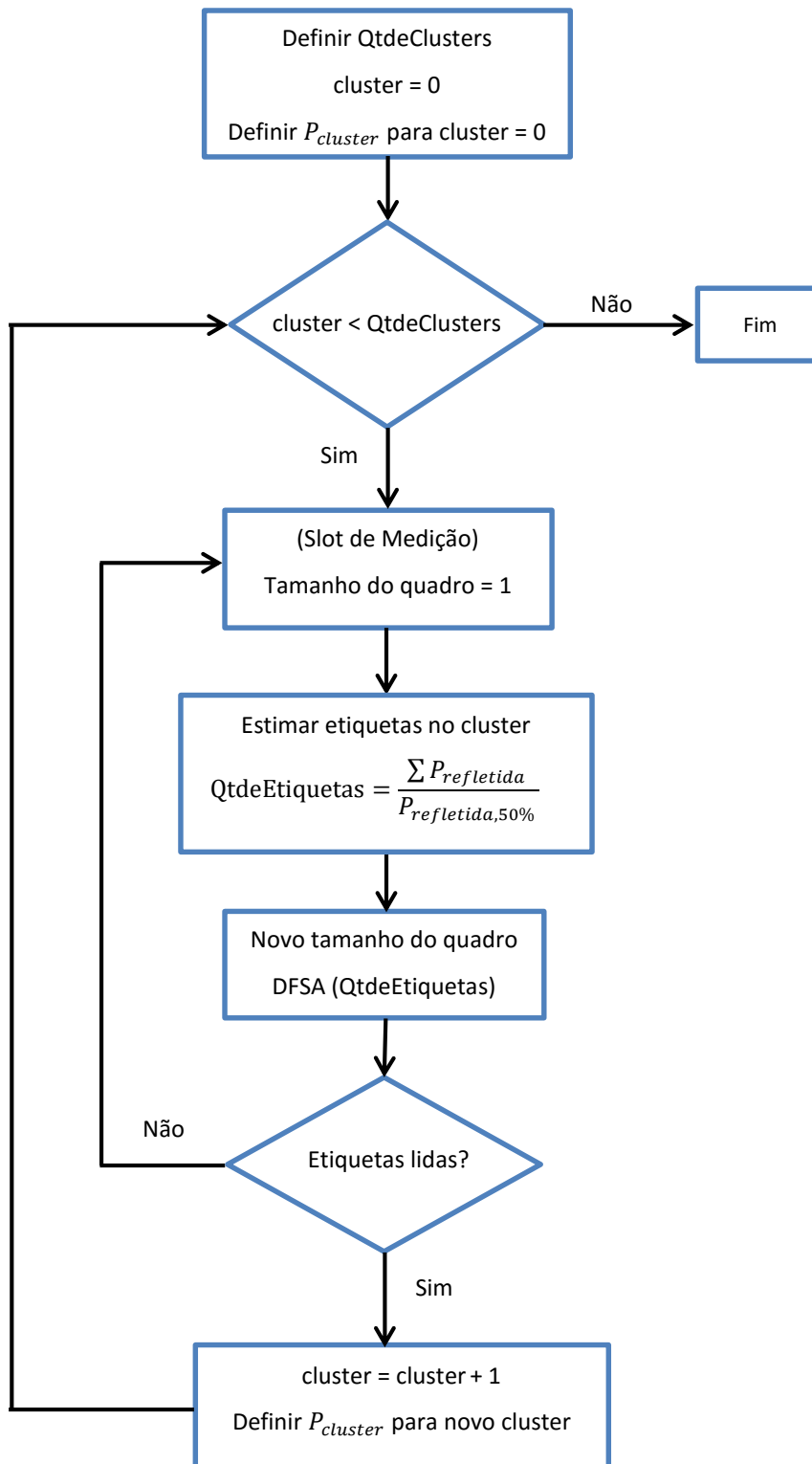


Figura 3.4 – Algoritmo Proposto.



## Capítulo 4

### Experimentos e Resultados

Neste capítulo é apresentada a avaliação de desempenho do mecanismo proposto de estimação por controle de potência. Esta avaliação é feita comparando-se o método proposto com diferentes algoritmos anticolisão e o algoritmo DFSA ideal. A quantidade de *slots* necessários para identificar todas as etiquetas de uma zona de interrogação é a medida principal da avaliação, mas outras medidas tais como: tempo de identificação das etiquetas, energia consumida pelo leitor, quantidade ótima de *clusters* para uma determinada quantidade de etiquetas e erros de estimativa também foram avaliadas.

#### 4.1 Simulador

Para a realização dos experimentos foi desenvolvido um simulador próprio escrito em linguagem C/C++ que permite a geração de resultados para os seguintes algoritmos: Mecanismo proposto, FSA 256, FSA 512, DFSA ideal e algoritmo *Q* padrão com diferentes quantidades de *clusters*. O simulador foi validado comparando-se os resultados obtidos com aqueles apresentados em artigos tais como no trabalho de Teng *et al.* (2010).

#### 4.2 Cenários de simulação

O cenário de simulação do mecanismo proposto é composto de um leitor localizado no centro de uma circunferência de raio 10m (zona de interrogação) e um grupo de etiquetas de acordo com a Figura 4.1. As posições das etiquetas na zona de interrogação foram geradas aleatoriamente de acordo com uma distribuição uniforme. Considera-se uma faixa de valores de 100 a 20.000 etiquetas para avaliar o desempenho em redes densas. A uma quantidade de *clusters* varia de 1 até 50. Para a geração dos resultados, calcula-se a média de 1.000 simulações com intervalo de confiança de 95%. O simulador considera que as etiquetas têm posições fixas durante todo o processo de simulação. E a frequência de operação é de 900 MHz de acordo com EPC Gen 2 UHF RFID.

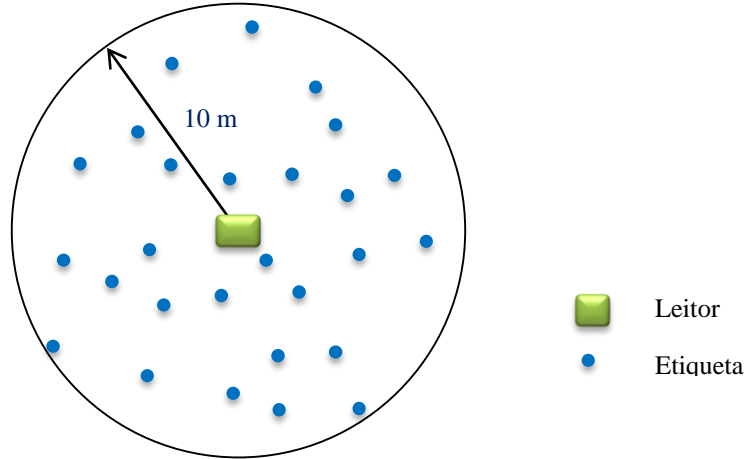


Figura 4.1 – Cenário de simulação.

As simulações foram realizadas considerando-se que a Equação (2.19) de *Friis*, com os parâmetros de antena calculados anteriormente, só é válida a partir de uma determinada de distância da antena do leitor (região de *Fraunhofer* ou de campo distante), ou seja, os *clusters* que estão localizados antes desta região devem ser interrogados utilizando qualquer algoritmo anticolisão. Nas simulações escolheu-se o algoritmo *Q*, pois é o algoritmo padrão da norma EPC Gen 2. O simulador define o início da região de *Fraunhofer* de acordo com a Equação (2.21). Supondo que a maior dimensão da antena de um leitor comercial seja de 40cm ( $D = 0,4m$ ) e que o comprimento de onda  $\lambda$  para 900 MHz é 0,33, então o valor aproximado será dado por:

$$R_f = \frac{2D^2}{\lambda} = \frac{2(0,4)^2}{0,33} = 0,96 \cong 1m$$

Para se determinar a quantidade de *clusters* que utilizam o algoritmo *Q* ( $QtdeClustersQ$ ), primeiro, calcula-se a largura de cada *cluster* ( $lc$ ) dividindo-se a largura da zona de interrogação ( $lz$ ) pela quantidade de *clusters* ( $QtdeClusters$ ) de acordo com a Equação (4.1).

$$lc = \frac{lz}{QtdeClusters} \quad (4.1)$$

Em seguida, divide-se a distância de *Fraunhofer* ( $R_f$ ) pela largura de cada *cluster* ( $lc$ ) conforme a Equação (4.2). Se o resultado não é um inteiro, arredonda-se para cima.

$$QtdeClustersQ = \frac{R_f}{lc} \quad (4.2)$$

A Tabela 4.1 mostra a quantidade de *clusters* que utilizam o algoritmo  $Q$  de acordo com a quantidade total de *clusters* da zona de interrogação. Por exemplo, caso a divisão da zona de interrogação ( $lz = 10m$ ) seja feita em 25 *clusters* ( $QtdeClusters$ ), cada *cluster* terá a largura de 0,4m ( $lc = 10m/25 \text{ clusters}$ ) e os *clusters* 0, 1 e 2 utilizam o algoritmo  $Q$ . Os outros 22 *clusters* utilizam o método de *slot* de medição.

Tabela 4.1 – *Clusters* que utilizam o algoritmo  $Q$ .  $R_f = 1m$ .

<b>Quantidade de <i>clusters</i></b>	<b><i>Clusters</i> que usam o algoritmo <math>Q</math></b>
[2,10]	0
[11,20]	0 e 1
[21,30]	0, 1 e 2
[31,40]	0, 1, 2 e 3
[41,50]	0, 1, 2, 3 e 4

### 4.3 Resultados Obtidos

O mecanismo proposto divide a zona de interrogação em *clusters*. Por isso, foram realizadas simulações para determinar qual a quantidade ótima de *clusters* em função da quantidade de etiquetas (Figura 4.2). Observa-se que, dependendo da faixa de etiquetas, é desejável que a quantidade de *clusters* esteja de acordo com o gráfico da Figura 4.2. Este gráfico foi elaborado baseado em gráficos similares ao mostrado na Figura 4.3, onde para cada quantidade de etiquetas, no caso 5.000 etiquetas, determinou-se a quantidade de *slots* em função da quantidade de *clusters* (5, 10, 15, 20, 30, 35, 40, 45, e 50).

No gráfico da Figura 4.2, percebe-se que, ao se aumentar a quantidade de etiquetas, deve-se aumentar a quantidade de *clusters* para que se tenha a menor quantidade de *slots* no

processo de identificação. Entretanto, como na prática não se sabe a quantidade exata de etiquetas, pode-se determinar uma aproximação de acordo com o tipo da aplicação.

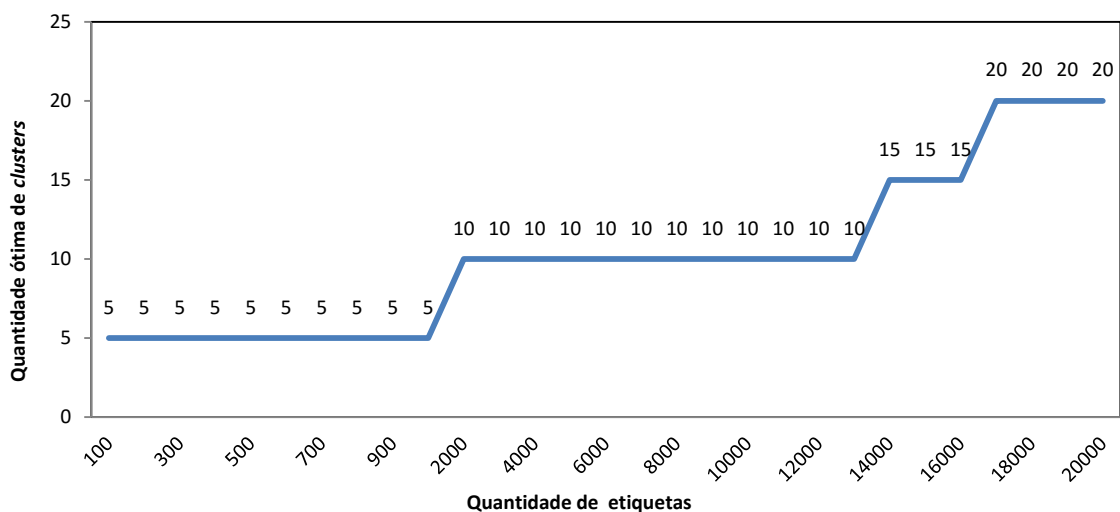


Figura 4.2 – Quantidade de etiquetas *versus* Quantidade ótima de *clusters*.

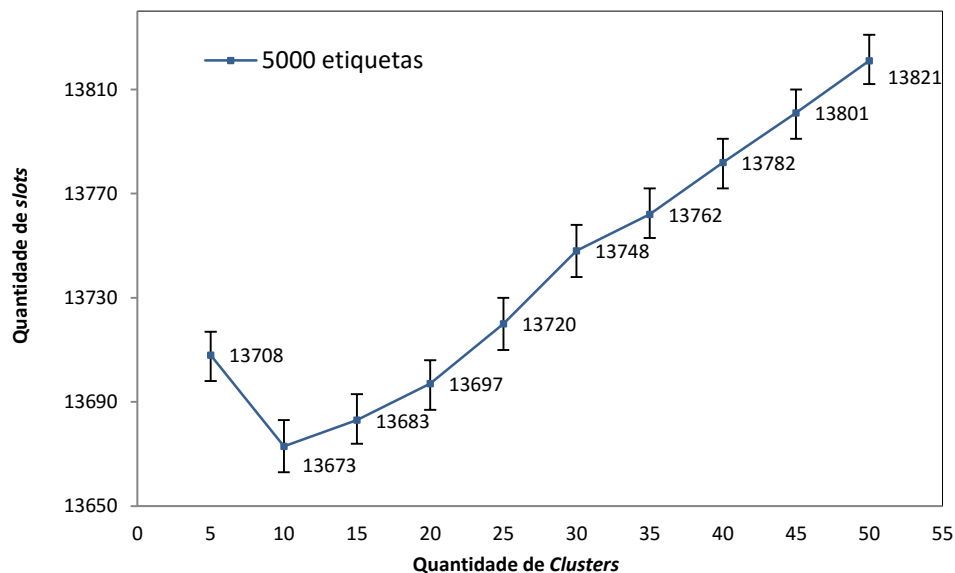


Figura 4.3 – Quantidade de clusters *versus* Quantidade de slots para 5.000 etiquetas.

Uma medida importante de avaliação do método proposto é o erro de estimativa por *cluster* da quantidade de etiquetas. Este erro é calculado de acordo com a Equação (4.3).

$$\text{Erro de estimativa por cluster}(\%) = \frac{\text{quant. etiq. estimadas} - \text{quant. etiquetas}}{\text{quant. etiquetas}} \times 100 \quad (4.3)$$

Os erros de estimativa para quatro diferentes quantidades de *clusters* (5, 10, 20 e 30) são apresentados na Figura 4.4 com as quantidades de etiquetas 500, 5.000, 15.000 e 20.000.

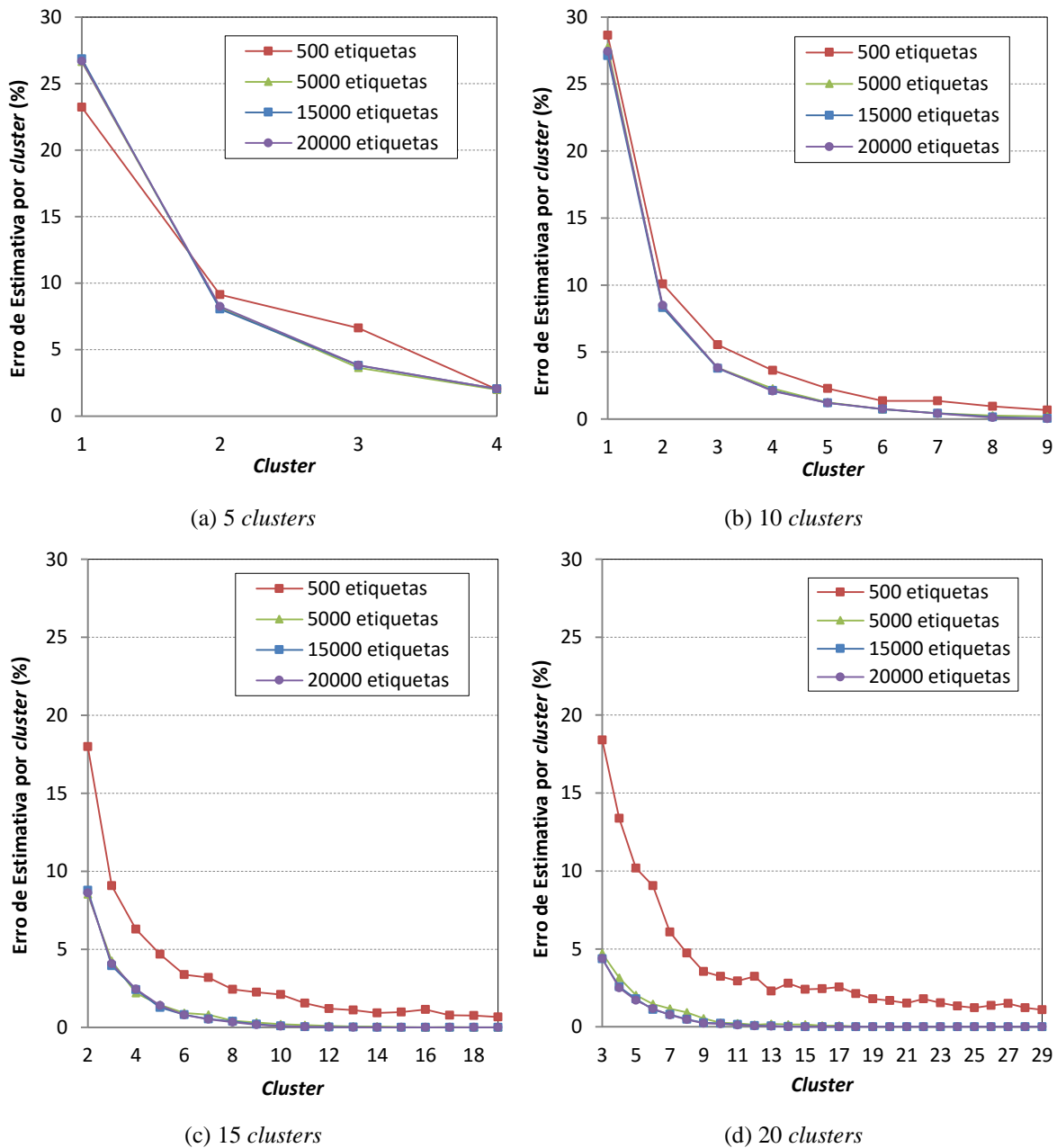


Figura 4.4 – Erros de Estimativa por *cluster* (a) 5 *clusters*, (b) 10 *clusters*, (c) 20 *clusters* e (d) 30 *clusters*.

Nota-se que os *clusters* iniciais apresentam um erro de estimativa maior do que o restante dos *clusters* em decorrência da potência refletida ser diretamente proporcional à  $\frac{1}{R^4}$ .

### 4.3.1 Quantidade de *slots*

Na Figura 4.5 é apresentado o gráfico da quantidade de *slots* em função da quantidade de etiquetas para os algoritmos DFSA ideal, FSA com quadro fixo de 256 *slots*, EDFSA, *Q* e o mecanismo proposto utilizando 20 *clusters*. A faixa varia de 100 a 1.000 etiquetas. Nesta faixa de etiquetas, o mecanismo proposto tem o melhor desempenho se a zona de interrogação for dividida em 5 *clusters*. Entretanto, a divisão em 20 *clusters* foi escolhida para que o mecanismo apresente o desempenho ótimo acima de 10.000 etiquetas.

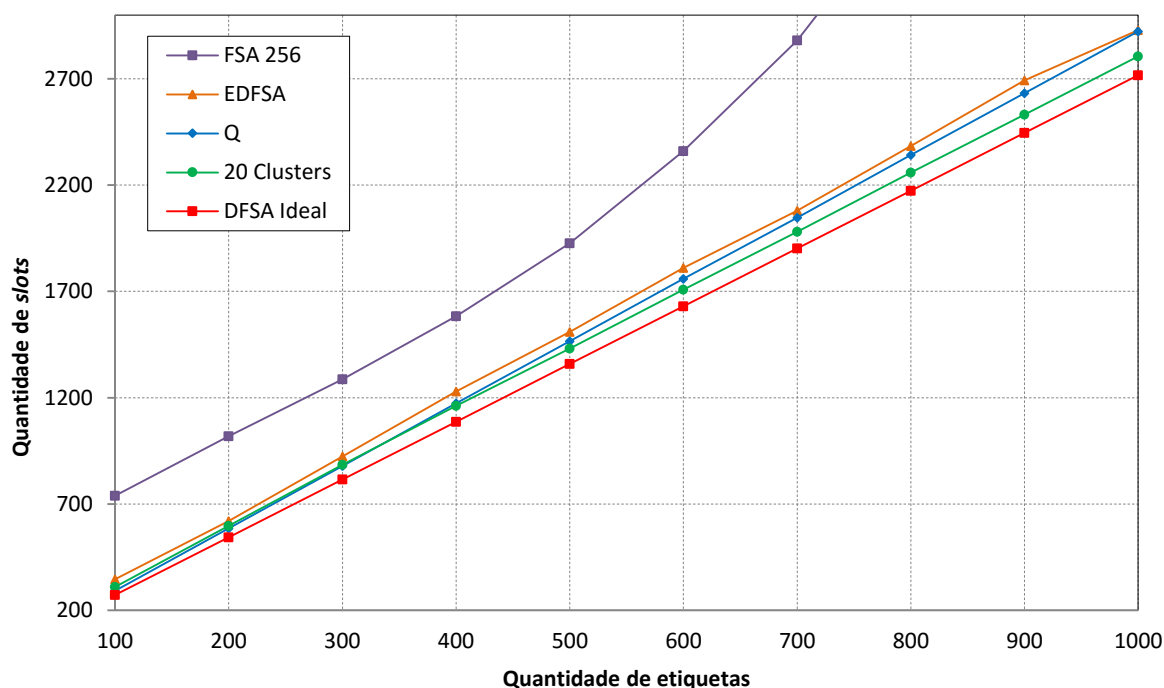


Figura 4.5 – Quantidade de etiquetas versus Quantidade de *slots*. (100 a 1.000 etiquetas).

Observa-se a menor quantidade de *slots* dos algoritmos *Q* e do mecanismo proposto (20 *clusters*) em relação aos algoritmos FSA 256 e EDFSA nesta faixa de etiquetas.

A Figura 4.6 mostra o desempenho para a faixa de 2.000 a 20.000 etiquetas e verifica-se que o mecanismo proposto (20 *clusters*) apresenta o desempenho muito próximo do ideal

(99 %) e uma melhoria de 6,5 % em relação ao algoritmo  $Q$ . Isso confirma a vantagem da divisão em *clusters* quando a quantidade de etiquetas é superior a 1.000.

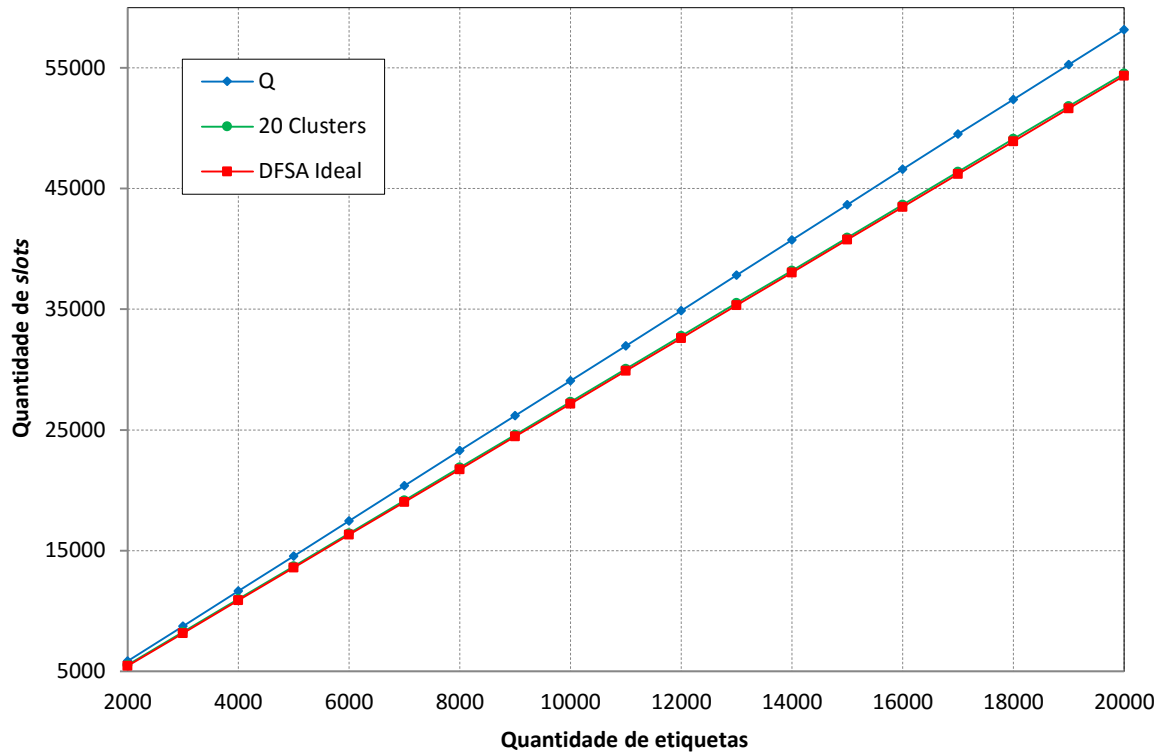


Figura 4.6 – Quantidade de etiquetas *versus* Quantidade de slots. (2.000 a 20.000 etiquetas).

Os gráficos das Figuras 4.5 e 4.6 foram plotados a partir da Tabela 4.2 que apresenta os resultados obtidos com a utilização do mecanismo proposto divididos em 10 clusters e 20 clusters. Para a simulação do algoritmo  $Q$  utilizou-se o parâmetro  $C = 0,2$  e a zona de interrogação não foi dividida em clusters.

Tabela 4.2 – Quantidade de *slots* para o algoritmo *Q* e mecanismo proposto.

Quantidade de Etiquetas	Quantidade de <i>slots</i>			
	DFSA Ideal	Q	10 <i>clusters</i>	20 <i>clusters</i>
100	272	291	301	310
200	543	585	579	597
300	815	879	856	884
400	1087	1174	1128	1161
500	1359	1466	1406	1431
600	1630	1760	1679	1709
700	1902	2047	1954	1981
800	2174	2342	2225	2259
900	2446	2633	2500	2532
1000	2717	2924	2769	2806
2000	5435	5842	5499	5533
3000	8152	8742	8230	8258
4000	10870	11660	10950	10987
5000	13587	14549	13688	13695
6000	16304	17446	16403	16425
7000	19022	20375	19125	19150
8000	21739	23296	21849	21884
9000	24457	26181	24571	24600
10000	27174	29076	27314	27322
11000	29891	31953	30038	30050
12000	32609	34867	32753	32775
13000	35326	37816	35500	35505
14000	38043	40741	38209	38200
15000	40761	43659	40925	40932
16000	43478	46592	43664	43663
17000	46196	49510	46385	46377
18000	48913	52368	49109	49107
19000	51630	55271	51819	51810
20000	54348	58156	54566	54534

### 4.3.2 Throughput

As medidas do *throughput* do algoritmo *Q* e do mecanismo proposto para diferentes *clusters* são apresentadas na Tabela 4.3. O *throughput* é baseado na Equação (2.12) e é definido de acordo com a Equação (4.4):



$$Throughput = \frac{\text{Quantidade de slots com sucesso}}{\text{Quantidade total de slots}} \times 100 \quad (4.4)$$

Tabela 4.3 – *Throughput* para algoritmo *Q* e mecanismo proposto.

Etiquetas	<i>Throughput</i> (%)					
	<i>Q</i>	<i>5 clusters</i>	<i>10 clusters</i>	<i>15 clusters</i>	<i>20 clusters</i>	<i>30 clusters</i>
100	34,25	33,67	33,90	32,15	31,35	30,40
200	34,07	35,03	34,54	33,78	33,39	32,21
300	34,01	35,63	35,13	34,44	34,13	33,48
400	34,07	35,94	35,56	35,09	34,54	33,87
500	34,15	35,95	35,69	35,21	34,97	34,25
600	34,13	36,10	35,76	35,44	35,17	34,62
700	34,23	36,18	35,86	35,62	35,25	34,77
800	34,16	36,17	35,91	35,73	35,40	35,03
900	34,18	36,19	36,04	35,77	35,59	35,10
1000	34,13	36,11	36,06	35,92	35,71	35,26
2000	34,23	36,34	36,34	36,29	36,15	35,87
3000	34,35	36,40	36,47	36,43	36,35	36,15
4000	34,28	36,39	36,51	36,50	36,43	36,29
5000	34,36	36,41	36,56	36,51	36,48	36,35
6000	34,38	36,44	36,58	36,55	36,52	36,44
7000	34,37	36,41	36,61	36,59	36,54	36,46
8000	34,32	36,45	36,61	36,60	36,56	36,49
9000	34,38	36,45	36,62	36,59	36,60	36,53
10000	34,42	36,44	36,60	36,61	36,61	36,53
11000	34,40	36,44	36,63	36,61	36,61	36,55
12000	34,42	36,45	36,64	36,63	36,63	36,57
13000	34,38	36,45	36,64	36,64	36,62	36,58
14000	34,37	36,46	36,65	36,64	36,64	36,58
15000	34,34	36,46	36,65	36,65	36,64	36,61
16000	34,33	36,44	36,64	36,66	36,66	36,61
17000	34,35	36,45	36,67	36,66	36,65	36,62
18000	34,37	36,46	36,65	36,66	36,66	36,63
19000	34,37	36,45	36,66	36,65	36,68	36,63
20000	34,38	36,42	36,65	36,65	36,67	36,64

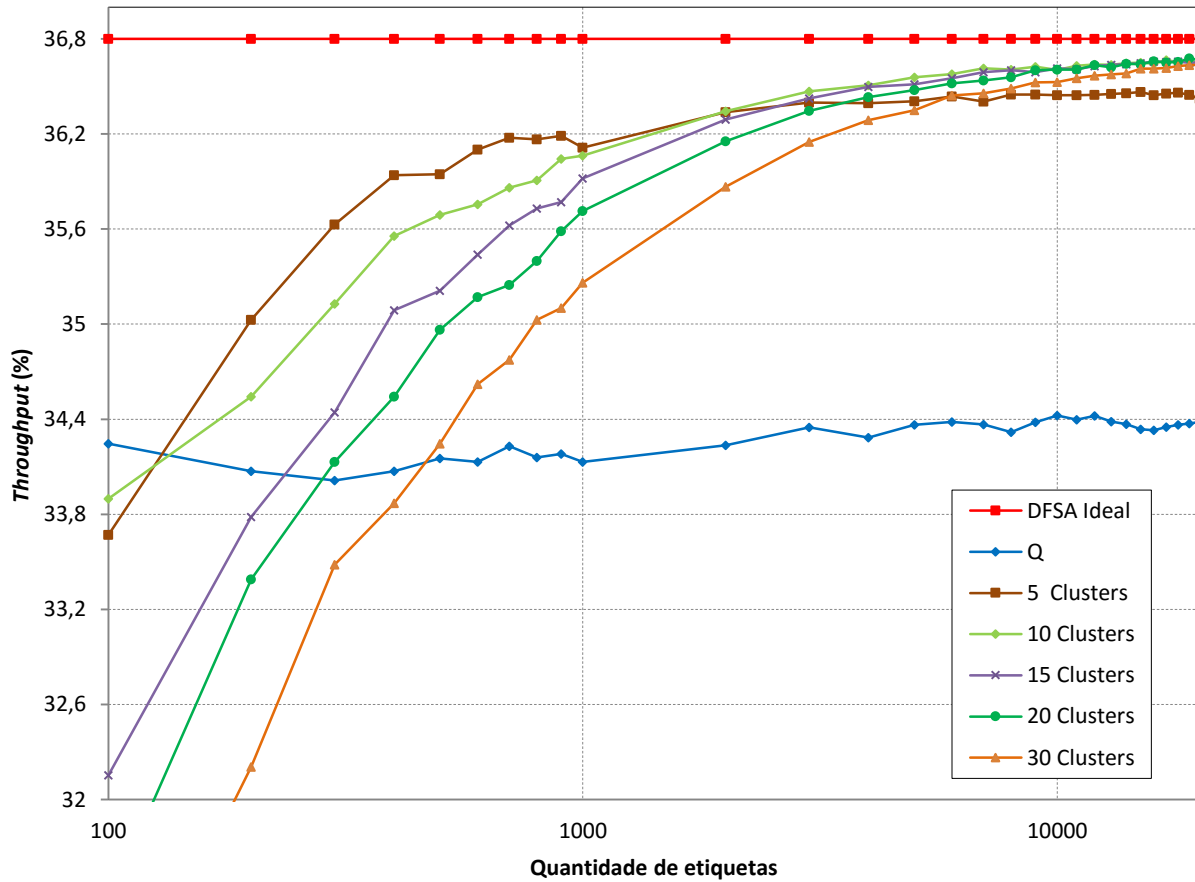


Figura 4.7 – Quantidade de etiquetas versus throughput (%).

A Figura 4.7 exibe o gráfico dos *throughputs* obtidos a partir da Tabela 4.3. Observa-se que algoritmo  $Q$  possui o *throughput* aproximadamente constante em função da quantidade de etiquetas, enquanto o mecanismo proposto apresenta um *throughput* dependente da quantidade de etiquetas a serem identificadas. Para uma faixa de 100 a 1.000 etiquetas, o *throughput* aumenta se a zona de interrogação for dividida em menos *clusters*. Isto acontece, pois, o mecanismo proposto utiliza mais *slots* de medição conforme a quantidade de *clusters* é aumentada. Por exemplo, se a zona de interrogação possuir 500 etiquetas, o *throughput* será maior se dividirmos em 5 *clusters* ao invés de 20 *clusters*. A partir de 1.000 etiquetas o comportamento tende a se inverter, ou seja, quanto maior a quantidade de etiquetas o *throughput* aumenta se a quantidade de *clusters* aumentar.

### 4.3.3 Tempo de identificação das etiquetas

A medida mais utilizada para avaliar o desempenho dos algoritmos anticisão é a quantidade de *slots*, pois facilita a comparação entre trabalhos de outros pesquisadores. Entretanto, muitos trabalhos utilizam o tempo de identificação das etiquetas por ser uma medida mais realista. A desvantagem da utilização do tempo de identificação decorre da dificuldade na comparação entre trabalhos distintos devido aos diferentes valores dos parâmetros, do padrão EPC Gen 2, para o cálculo dos tempos dos *slots* (com sucesso, com colisão e vazios).

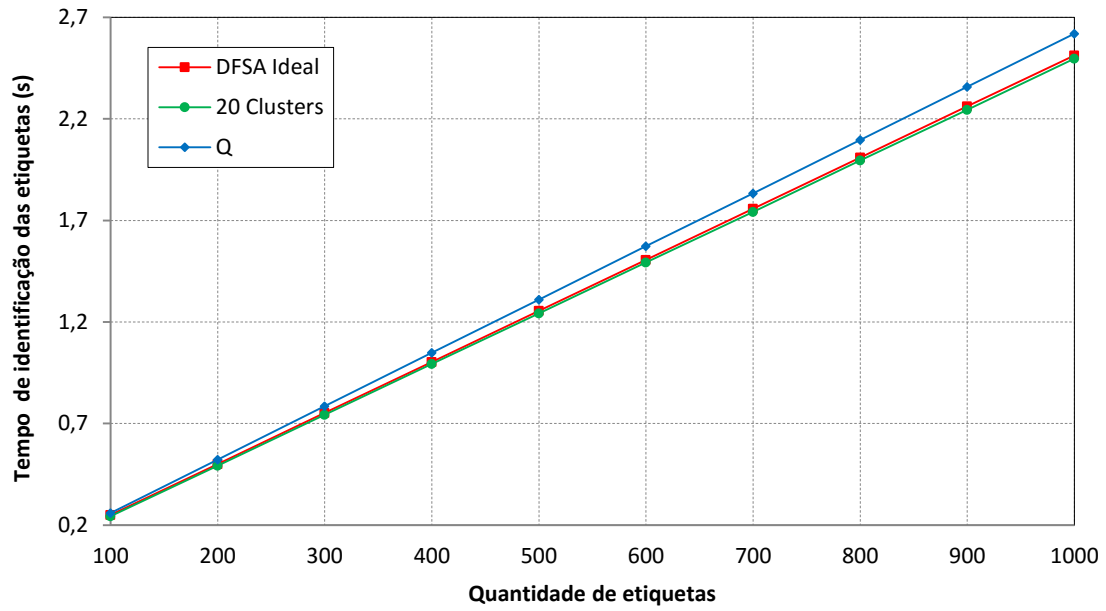
Na Tabela 4.4, são apresentados os valores do tempo de identificação das etiquetas em função da quantidade de etiquetas. Para a simulação do tempo de identificação das etiquetas é necessário definir os valores para os parâmetros do padrão EPC Gen2. Algumas considerações foram feitas baseadas no trabalho de Chen (2012), que define a taxa de transmissão em 80kbps, e a partir deste parâmetro foram calculados os tempos do *slot* com sucesso (1,912ms), do *slot* com colisão (0,475ms) e do *slot* vazio (0,262ms). Os resultados para o DFSA ideal também foram obtidos por meio de simulação.

Na Figura 4.8 são mostrados os gráficos plotados a partir da Tabela 4.4. Observa-se na Figura 4.8 (a) o comportamento do tempo de identificação para a faixa de 100 a 1.000 etiquetas.

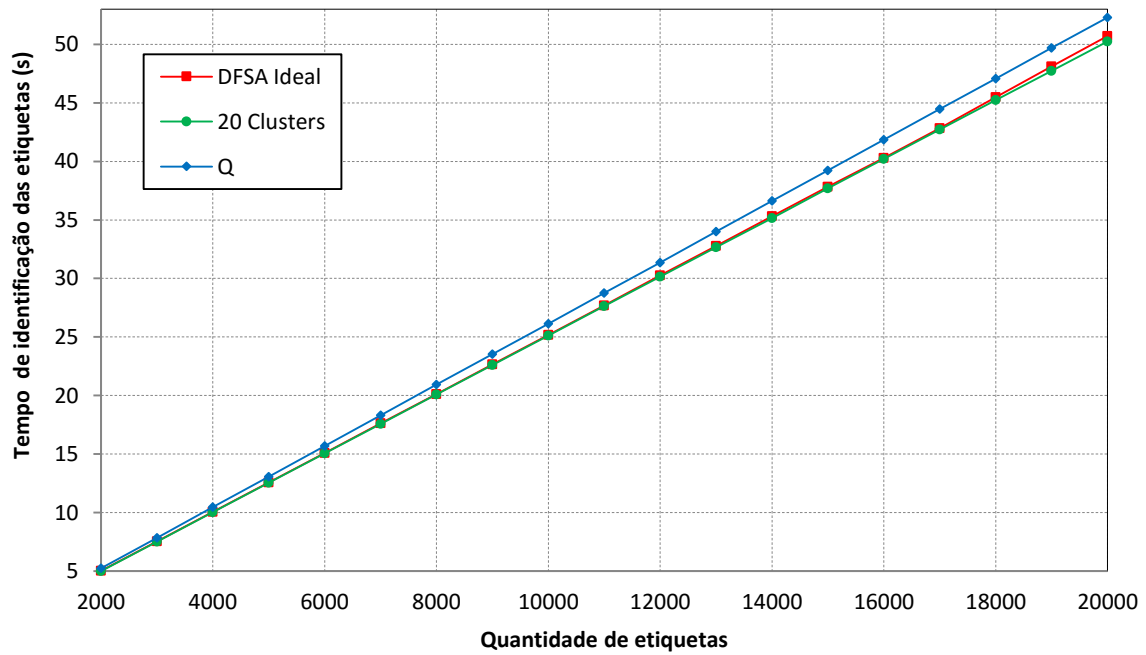
Pelos resultados apresentados, percebe-se que o tempo de identificação das etiquetas teve uma redução em torno de 0,5 % em comparação ao DFSA ideal. A aparente contradição se deve ao fato de que o algoritmo DFSA ideal foi simulado sem a divisão da zona de interrogação em *clusters*. Em comparação ao algoritmo  $Q$  o tempo de identificação do mecanismo proposto teve uma redução em torno de 4%.

Tabela 4.4 – Tempo de Identificação para algoritmo  $Q$ , mecanismo proposto e DFSA ideal.

Quantidade de Etiquetas	Tempo de Identificação (s)		
	DFSA Ideal	20 clusters	$Q$
100	0,25	0,24	0,26
200	0,50	0,49	0,52
300	0,75	0,74	0,79
400	1,00	0,99	1,05
500	1,26	1,24	1,31
600	1,51	1,49	1,57
700	1,76	1,74	1,83
800	2,01	2,00	2,10
900	2,26	2,25	2,36
1000	2,51	2,50	2,62
2000	5,03	5,01	5,24
3000	7,54	7,52	7,85
4000	10,06	10,03	10,47
5000	12,58	12,54	13,08
6000	15,10	15,05	15,69
7000	17,62	17,57	18,31
8000	20,12	20,09	20,93
9000	22,66	22,60	23,54
10000	25,18	25,11	26,15
11000	27,68	27,63	28,75
12000	30,25	30,14	31,37
13000	32,78	32,66	33,99
14000	35,32	35,16	36,62
15000	37,82	37,68	39,23
16000	40,28	40,20	41,86
17000	42,83	42,71	44,48
18000	45,48	45,23	47,07
19000	48,11	47,73	49,68
20000	50,70	50,25	52,29



(a) Faixa de 100 a 1.000 etiquetas.



(b) Faixa de 2.000 a 20.000 etiquetas

Figura 4.8 – Quantidade de etiquetas *versus* Tempo de identificação.

#### 4.3.4 Energia consumida pelo leitor

A Tabela 4.5 e a Figura 4.9 apresentam os resultados da energia consumida pelo leitor em função da quantidade de etiquetas. Para se obter o consumo de energia total do leitor, utilizou-se a Equação (4.5) onde  $P_{total}$  é a potência fornecida pelo leitor (XU *et al.*, 2011), durante o tempo  $t_{total}$  de identificação de todas as etiquetas.

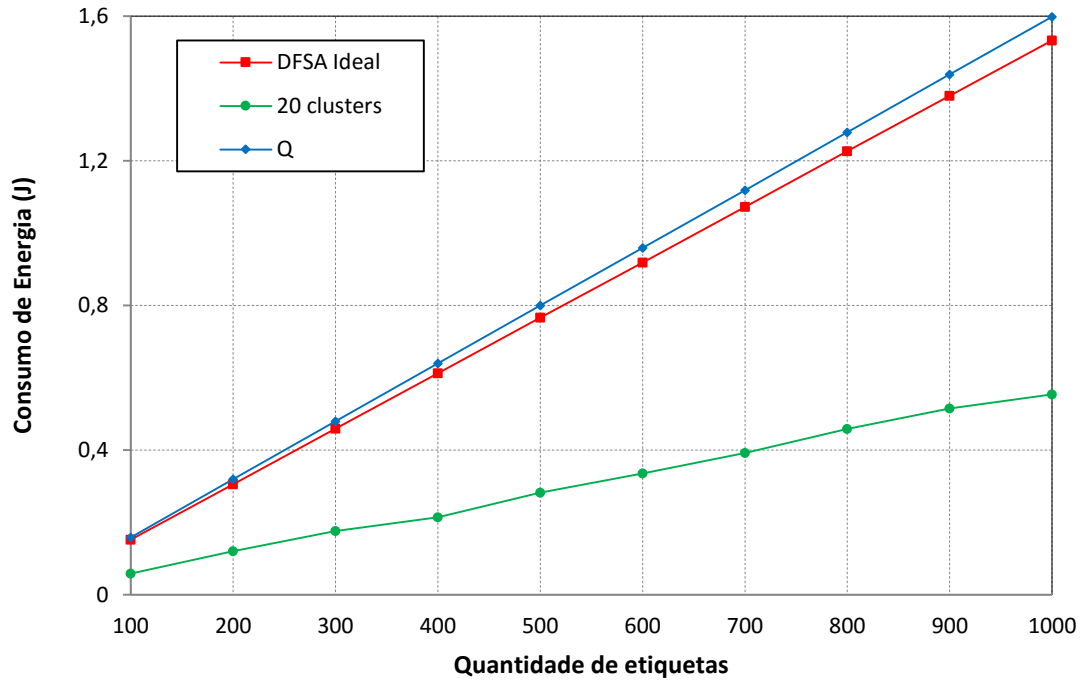
$$E_{total} = P_{total} \cdot t_{total} \quad (4.5)$$

No mecanismo proposto, a energia total fornecida foi calculada somando-se a energia consumida em cada *cluster* de acordo com a Equação (4.6).

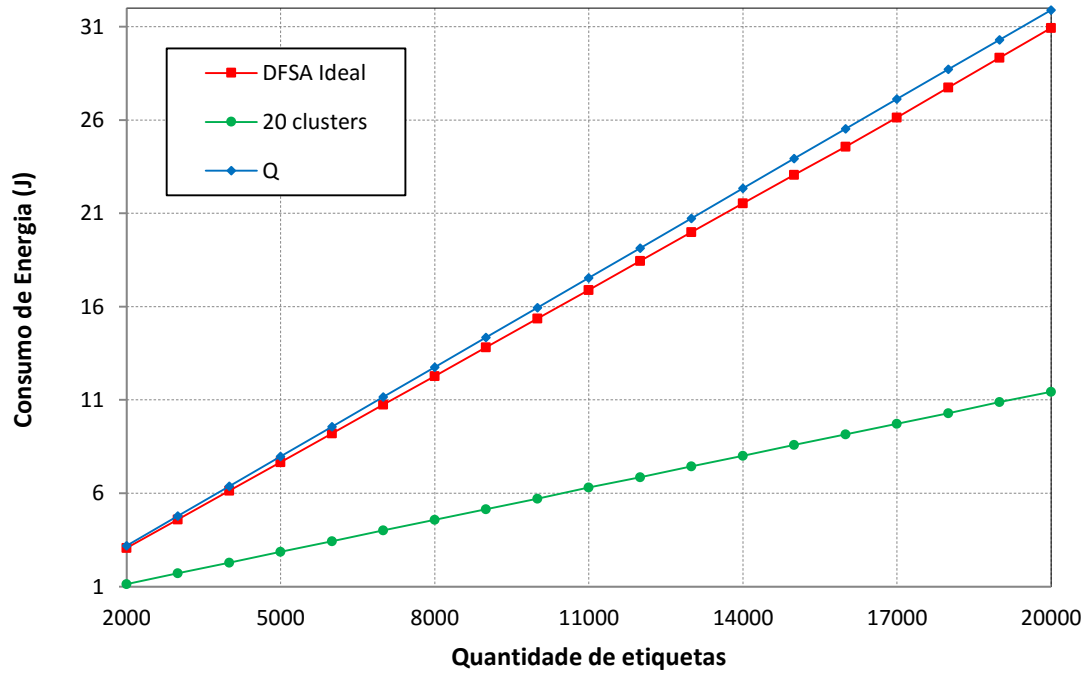
$$E_{total} = \sum P_{cluster} \cdot t_{cluster} \quad (4.6)$$

Tabela 4.5 – Consumo de energia do leitor para algoritmo  $Q$ , mecanismo proposto e DFSA ideal.

Quantidade de etiquetas	Energia consumida pelo leitor (J)		
	DFSA Ideal	20 clusters	$Q$
100	0,15	0,06	0,16
200	0,31	0,12	0,32
300	0,46	0,18	0,48
400	0,61	0,21	0,64
500	0,77	0,28	0,80
600	0,92	0,34	0,96
700	1,07	0,39	1,12
800	1,23	0,46	1,28
900	1,38	0,51	1,44
1000	1,53	0,55	1,60
2000	3,07	1,14	3,20
3000	4,60	1,71	4,79
4000	6,14	2,28	6,39
5000	7,67	2,87	7,98
6000	9,21	3,43	9,57
7000	10,75	4,01	11,17
8000	12,27	4,58	12,77
9000	13,82	5,15	14,35
10000	15,36	5,72	15,95
11000	16,88	6,30	17,53
12000	18,45	6,86	19,13
13000	19,99	7,44	20,73
14000	21,54	8,00	22,33
15000	23,07	8,59	23,93
16000	24,57	9,16	25,53
17000	26,12	9,73	27,13
18000	27,74	10,30	28,71
19000	29,34	10,88	30,30
20000	30,93	11,45	31,89



(a) Faixa de 100 a 1.000 etiquetas



(b) Faixa de 2.000 a 20.000 etiquetas

Figura 4.9 – Quantidade de etiquetas *versus* Consumo de energia do leitor.



É possível notar uma redução de 63% no consumo de energia do leitor em relação ao algoritmo DFSA ideal. A redução em comparação ao algoritmo  $Q$  foi de 65%. Observa-se que a divisão por *clusters* reduz bastante o consumo de energia do leitor, pois o transmissor não tem necessidade de fornecer sempre a máxima potência para atingir toda a zona de interrogação e aguardar o algoritmo anticollisão interrogar todas as etiquetas. O leitor aumenta a potência gradativamente, o que permite a identificação das etiquetas dos *clusters* iniciais com a potência reduzida.

## Capítulo 5

### Conclusões e Trabalhos Futuros

#### 5.1 Conclusões

Este trabalho realizou estudos sobre os algoritmos anticisão baseados em ALOHA (DFSA) e propôs uma nova abordagem para estimar a quantidade de etiquetas em redes RFID.

Os algoritmos DFSA precisam estimar, a partir do quadro atual lido, o tamanho do próximo quadro. Mostrou-se que, para obter a melhor eficiência, o tamanho do próximo quadro enviado às etiquetas deve ser igual à quantidade de etiquetas ainda não lidas. Os estimadores tradicionais baseiam-se na contagem da quantidade de *slots* com sucesso, de *slots* com colisão e de *slots* vazios. Estes estimadores apresentam algumas desvantagens quanto à acurácia da estimativa ou quanto ao custo computacional envolvido, dada a complexidade de alguns estimadores probabilísticos. Outra observação importante decorre do fato de que os estimadores apresentam perda de desempenho quando a quantidade de etiquetas aumenta (acima de 1.000), ou seja, em ambientes de alta densidade de etiquetas. Isto ocorre porque a quantidade de colisões aumenta consideravelmente.

No Capítulo 3, foi apresentado o mecanismo de controle de potência para aumentar o desempenho dos algoritmos DFSA em ambientes de alta densidade. O mecanismo proposto baseou-se na divisão da zona de interrogação em *clusters*, com o objetivo de diminuir a quantidade de colisões, e na medição da intensidade do sinal retornado pelas etiquetas. A medida da intensidade dos sinais refletidos pelas etiquetas pode ser realizada de modo relativamente simples.

No Capítulo 4, os resultados das simulações foram apresentados e constatou-se que a quantidade de *slots*, para identificar todas as etiquetas pelo mecanismo proposto, apresentou desempenho de 99% do algoritmo DFSA ideal em redes densas, onde há grande quantidade de etiquetas. Em relação ao algoritmo  $Q$ , percebeu-se a melhoria de 6,5% no desempenho.

Pelos resultados apresentados, percebeu-se que o tempo de identificação teve uma redução em torno de 0,5% em relação ao algoritmo DFSA ideal e de 4% em comparação ao algoritmo  $Q$ .

A energia consumida pelo leitor apresentou grande redução, em torno de 63% comparada ao algoritmo DFSA ideal e de 65% em relação ao algoritmo  $Q$ . Esta redução decorre da divisão da área de interrogação em *clusters*, pois o leitor aumenta a potência fornecida de acordo com cada *cluster* lido. A redução na energia consumida permite que aplicações com leitores portáteis apresentem uma melhora na autonomia de suas baterias.

## 5.2 Trabalhos Futuros

Nesta seção serão apresentadas algumas sugestões de trabalhos futuros, que poderão dar continuidade aos experimentos realizados para esta dissertação.

Nos experimentos realizados, as etiquetas foram consideradas tendo posições fixas. Em cenários onde as etiquetas são móveis, pode-se avaliar o mecanismo proposto quanto aos erros de estimativa por *cluster*, simular cenários críticos, tais como um grupo de etiquetas se deslocando em direção ao leitor ou se afastando do leitor, avaliar o impacto da velocidade das etiquetas e propor alterações que minimizem os erros de estimativas.

Pretende-se aprimorar o modelo de propagação entre o leitor e as etiquetas de modo a simular com mais precisão o comportamento real das etiquetas. Pode-se fazer a simulação do posicionamento angular das etiquetas e avaliar o impacto na potência refletida no medidor de potência RSSI. Deve-se simular a região de fronteira entre dois *clusters*, pois esta área apresenta pequenas variações em seu tamanho com o tempo. Uma etiqueta presente nesta área pode pertencer a um *cluster* em um determinado momento e a outro *cluster* no instante seguinte. O mecanismo deve se adequar a estas variações para que não haja perda de precisão na estimativa.

Outra sugestão, seria avaliar novos cenários com diferentes distribuições de etiquetas e aprimorar o mecanismo para se adaptar dinamicamente a estes novos cenários. O leitor poderia manter um histórico das distribuições e quantidades de etiquetas e configurar automaticamente a quantidade e/ou a largura dos *clusters*.

## Referências Bibliográficas

- ALI, K., HASSANEIN, H. and TAHA, A-E. “RFID Anti-collision Protocol for Dense Passive Tag Environments”. In: Proceedings of the International Conference on Local Computer Networks, pp. 819-824. Dublin, 2007.
- ALOTAIBI, M., POSTULA, A. and PORTMANN, M. “A Signal Strength based Tag estimation technique for RFID systems”. In: Proceedings of the International Conference on RFID-Technology and Applications (RFID-TA), pp. 251-256, Guangzhou-China 2010.
- ALSALIH, W., ALI, K. and HASSANEIN, H. “A Power Control Technique for anti-collision schemes in RFID systems”. Journal of Computer Networks, 57, pp. 1991-2003, Elsevier, 2013.
- BOLIC, M., SIMPLOT-RIY, D., STOJMENOVIC, I. “RFID Systems - Research Trends and Challenges”. John Wiley & Sons Ltd. West Sussex, United Kingdom, 2010.
- BUENO-DELGADO, M.V., VALES-ALONSO, J. and GONZALEZ-CASTAÑO, F.J. “Analysis of DFSA Anti-collision Protocols in passive RFID environments”. In: Proceedings of the Industrial Electronics, pp. 2610 – 2617. Porto, 2009.
- CHEN, W. and LIN, G.-H. “An Efficient Anti-Collision Method for RFID system”. Journal IEICE Transactions Communication. E89 (B), pp. 3386–3392, 2006.
- CHEN, W. “A new RFID Anti-collision algorithms for the EPCglobal UHF Class-I Generation-2 standard”. In: Proceedings of the International Conference on Ubiquitous Intelligence and Computing and 9<sup>th</sup> International Conference on Autonomic and Trusted Computing, pp. 811-815. Fukuoka, 2012.
- DHAKAL, S. and SHIN, S. “A sequential reading strategy to improve the performance of RFID anti-collision algorithm in dense tag environments”. In: Proceedings of the International Conference on Ubiquitous and Future Networks (ICUFN), pp. 531-536. Da Nang, 2013.

- DOBKIN, D. M. “The RF in RFID: Passive UHF RFID in Practice”. Elsevier Inc. Newnes, 2012.
- EOM, J.-B. and LEE, T. J. “Accurate Tag Estimation for Dynamic Framed-Slotted ALOHA in RFID Systems”. IEEE Communications Letters, vol. 14, n° 1, 2010.
- EPC Radio-Frequency Identity Protocols Class-1 Generation 2 UHF RFID Protocol for Communications at 860 MHz 960 MHz, Version 2.0.0, EPCglobal, 2013.
- FINKENZELLER, K. “RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification”. John Wiley & Sons, Inc. New York, NY, USA, 2010.
- FRIIS, H. T. “A Note on a Simple Transmission Formula”. In: Proceedings of the IRE and Waves and Electrons May, n° 1, pp. 254–256, 1946.
- GLOVER, B and BHATT, H. “RFID. Essentials”. O’Reilly Media Inc. Sebastopol, CA, USA, 2006.
- GUILAN, L. and GUOCHAO, Z. “An improved anti-collision algorithm in RFID system of Internet of Things”. In: Proceedings of the Conference on Multimedia Technology (ICMT), 10. pp. 1–4. Ningbo-China. 2010.
- JIA, X., FENG, Q. and YU, L. “Stability analysis of an efficient anti-collision protocol for RFID tag identification”. IEEE Journal on Transactions on Communications, vol. 60, n° 8, pp. 2285 –2294, 2012.
- KLAIR, D. K., CHIN, K.-W. and RAAD, R. “A Survey and Tutorial of RFID Anti-collision Protocols”. IEEE Journal on Communications Surveys & Tutorials, vol. 12, N° 3, 2010.
- LEE, S. and LEE, C. “An Enhanced Dynamic Framed Slotted ALOHA Algorithm for RFID Tag”. In: Proceedings of the Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous), pp. 1–7, 2005.

- LEE, D., KIM, K. and LEE, W. “Q+ Algorithm: An Enhanced RFID Tag Collision Arbitration Algorithm”. In: Proceedings of the International Conference Ubiquitous Intelligence and Computing. Volume 4611, pp. 23-32. Hong Kong-China, 2007.
- SCHOOTE, F. C. “Dynamic frame length ALOHA”. IEEE Transactions on Communications, COM-31(4), pp. 565–568, 1983.
- SU, W., ALCHAZIDIS N. and HA, T. “Multiple RFID Tags Access Algorithm”. IEEE Transactions on Mobile Computing, vol. 9, n° 2, pp. 174-187, 2010.
- TENG, J., XUAN, X. and BAI, Y. “A Fast Q Algorithm Based on EPC Generation-2 RFID Protocol”. In: Proceedings of the International Conference on Computational Intelligence and Software Engineering, pp. 1–4. Chengdu, 2010.
- VOGT, H. “Efficient object identification with passive RFID tags”. In: Proceedings of the International Conference on Pervasive Computing, pp. 98–113. Zurich, 2002.
- XU X., GU, L., J. WANG, and XING, G. “Read More with Less: An Adaptive Approach to Energy-Efficient RFID Systems”. IEEE Journal on Select Areas and Communications, vol. 29, n° 8, pp. 1684-1697, 2011.
- ZHONG, W., CHEN, J., WU, L. and PAN, M. “The application of ALOHA algorithm to anti-collision of RFID tags”. In: Proceedings of the International Conference on Measurement, Information and Control. Vol. 1, pp. 717–720. Harbin-China, 2012.