



Universidade Federal de Pernambuco

Centro de Informatica

Graduação em Ciência da Computação

Um estudo sobre Protocolos Anti-Colisão para Sistemas RFID

Caio Franco Mc Intyre

Trabalho de Graduação

Recife,

14 de Dezembro de 2010

Universidade Federal de Pernambuco

Centro de Informatica

Caio Franco Mc Intyre

Um estudo sobre Protocolos Anti-Colisão para Sistemas RFID

Trabalho de Graduação apresentado ao Centro de Informática da Universidade Federal de Pernambuco por Caio Franco Mc Intyre como um requisito parcial para se obter o grau de bacharel em Ciência da Computação.

Orientador: Paulo André da Silva Gonçalves

Recife,

14 de Dezembro de 2010

Dedico este trabalho à minha família, principalmente aos meus pais e irmãos que me acompanharam durante toda essa jornada. À minha namorada que me ajudou muito em momentos difíceis e delicados da minha vida, e também aos meus colegas de universidade que me deram a força e ajuda necessária nesses momentos para poder seguir em frente com tranquilidade. Por fim agradeço ao meu orientador, o Prof. Paulo Gonçalves, que sempre me ajudou a desenvolver o meu potencial e colaborou muito para o meu desenvolvimento na vida acadêmica.

Agradecimentos

Primeiramente gostaria de agradecer o Prof. Paulo Gonçalves por desde o começo da minha vida acadêmica ter aceitado trabalhar junto comigo e por ter aberto os meus olhos para o mundo de redes e de sistemas RFID.

Em seguida, gostaria de agradecer aos meus pais que sempre acompanharam os meus passos e me incentivaram a ir mais longe. Eles me ensinaram o valor dos estudos e da determinação na vida de uma pessoa e o quão longe ela pode chegar com esses dois simples pilares.

Agradeço aos meus irmãos que muitas vezes mesmo sem entender o que eu estava fazendo ou até sem querer entender nunca se negaram a ajudar e dar suporte nos momentos em que precisei.

A minha namorada que me ajudou a sempre acreditar que um amanhã melhor virá e me ajudou a superar muito obstáculos que apareceram pelo caminho.

Aos meus colegas de faculdade e amigos que me ajudaram no momento da vida em que mais precisei de ajuda. Precisar ficar em casa de repouso enquanto o semestre estava rolando e ainda cursar normalmente a universidade não foi uma tarefa fácil e certamente não teria sido possível sem vocês. Vocês se dispuseram a tirar minhas dúvidas e trabalhar comigo mesmo que aquilo significasse perder o pouco do tempo livre que vocês tinham. Terminar a faculdade tão bem não teria sido possível sem vocês.

Por fim eu agradeço aos colegas do grupo de estudos do Prof. Paulo que me ajudaram muito no mundo da pesquisa científica e que sem vocês a realização desse trabalho não seria possível.

Resumo

O mundo da tecnologia da informação esta em constante expansão e seu crescimento esta cada vez maior. Dentro desse cenario é visivel o crescimento das aplicações que utilizam sistemas RFID para solucionar os problemas de comunicação a distância e simultânea entre objetos.

A idéia deste trabalho é ajudar na evolução dos sistemas RFID como um todo através da avaliação de desempenho de alguns protocolos anti-colisão, tentando entender os motivos pelos quais eles obtém tais resultados e tentando mostrar os pontos fracos e fortes de cada um para que eles possam ser trabalhados e melhorado.

Este trabalho faz isso estudando o estado da arte dos protocolos anti-colisão baseados em ALOHA para sistemas RFID. Também realiza um estudo aprofundado nos métodos de estimativa de tamanho de frame utilizados nos protocolos do tipo Dynamic Framed-Slotted ALOHA. Finalmente, ele apresenta uma avaliação de desempenho extensa dos protocolos tanto no ambiente ideal teorico quanto num ambiente realistico. Os protocolos estudados são Lower Bound, Schoute, Vogt e do Eom-Lee, um dos mais recentes e mais relevantes na literatura acadêmica.

Palavras-Chave: RFID, Anti-Colisão, Dynamic Frame-Slotted ALOHA, Capture Effect, Lower Bound, Schoute, Vogt ,Eom-Lee

Abstract

The information technology is in continuous expansion and its growth is getting bigger by the day. Within this scenario, the growth of RFID systems that solve the problems of applications that need objects with distance and simultaneous communication is visible.

The main idea of this work is to help on the evolution of RFID systems as whole by making performance evaluations of some anti-collision protocols, trying to understand the reasons why they achieve some results. The goal is to show their weaknesses and their strengths so that we may improve them.

This work studies the basic concepts of ALOHA anti-collision protocols for RFID systems. It also makes a profound study on the frame calculation methods of the Dynamic Framed-Slotted ALOHA protocols. Finally, it presents a performance evaluation in ideal theoretical and realistic environments. The studied protocols are Lower Bound, Schoute, Vogt and Eom-Lee, one of the most recent and relevant protocols in the academic literature.

Sumário

Sumário	7
1. Introdução	9
1.1. Motivação.....	9
1.2. Objetivo	9
1.3. Estrutura.....	10
2. Conceitos Básicos	11
2.1. Sistemas RFID	11
2.1.1. Leitores RFID	11
2.1.2. Etiquetas RFID	12
2.2. Colisões em Sistemas RFID.....	12
2.3. Protocolos Anti-Colisão	13
2.3.1. Protocolos Baseados em Árvore	14
2.3.2. Protocolos ALOHA	15
2.4. Protocolos DFSA Estudados	18
2.4.1. Lower Bound	18
2.4.2. Schoute.....	18
2.4.3. Vogt	19
2.4.4. Eom-Lee.....	21
2.5. <i>Capture Effect</i>	23
2.5.1. Calculando o <i>Capture Effect</i>	24
3. Análise	25
3.1. Validação do Simulador.....	25
3.1.1. Informações Avaliadas	25
3.1.2. Validação do DFSA.....	25
3.1.3. Validação do <i>Capture Effect</i>	26
3.2. Simulações.....	28
3.2.1. Avaliação da Performance dos Protocolos sem <i>Capture Effect</i>	29

3.2.2.	Avaliação do Comportamento dos Protocolos Com <i>Capture Effect</i>	31
3.2.3.	Avaliação do Impacto do <i>Capture Effect</i> na <i>Performance</i>	33
3.3.	Considerações	36
3.3.1.	<i>Performance</i> de Reconhecimento.....	36
3.3.2.	<i>Capture Effect</i>	36
4.	Conclusão	38
5.	Referências.....	39
Anexos.....		41
Anexo A – Resultado das Simulações – <i>Slots x Tags</i> sem o efeito de captura.....		41
Anexo B – Resultado das Simulações – <i>Slots x Tags</i> com o efeito de captura.....		43
Anexo C – Resultado das Simulações – Eficiência Com <i>Capture Effect</i> Vs Eficiência Sem <i>Capture Effect</i>		45

1. Introdução

1.1.Motivação

Sistemas RFID, *Radio Frequency Identification*, são sistemas de identificação de objetos que funcionam por rádio frequência. Considerado o sucessor do código de barras [1] sistemas RFID podem ser utilizados em diversas aplicações e são muito úteis para a melhora das aplicações do dia-dia. Exemplos de aplicações vão desde a utilização de etiquetas de produtos em um supermercado até aplicações que monitoram o desenrolar de um jogo de baralho, onde cada carta tem uma etiqueta de identificação e o jogo é monitorado por uma aplicação [2].

Aplicações baseadas em sistemas RFID são cada vez mais usadas. Aplicações de rastreamento e localização de objetos estão sendo usadas no gerenciamento de cadeias de fornecimento, *Supply Chain Management*, na automação industrial [3] e até em aplicações que reconhecem quais objetos uma mulher carrega na bolsa, quais ela tirou da bolsa e quando[4].

Com uma previsão da ABI Research de atingir a marca de 9,7 bilhões de dólares em 2013 [5], o mercado das etiquetas de identificação por radio frequência, está cada vez mais atrativo. Esse crescimento de mercado reflete a necessidade das pessoas e empresas de simplificar cada vez mais seu dia-dia utilizando aplicações que permitam identificação à distância.

Pesquisas para desenvolver cada vez mais esses sistemas estão em andamento neste momento no mundo inteiro. Desde pesquisas para melhorar o desempenho dos sistemas RFID, permitindo a criação de sistemas mais complexos, até pesquisas para usar as etiquetas como sistemas de localização [6].

Uma das áreas de pesquisa para aumentar a performance dos sistemas é a que tem por objetivo diminuir a quantidade de colisões. Colisões são interferências que uma etiqueta pode causar na outra no momento de comunicação com o leitor das etiquetas. Esses protocolos que tentam evitar essas interferências são chamados de protocolos anti-colisão já que tentam tratar a colisão de informações provenientes das diferentes etiquetas.

1.2.Objetivo

Existem diversos protocolos anti-colisão e o objetivo deste trabalho é estudar esses protocolos anti-colisão, focando no recém criado Eom-Lee mostrando como seu desempenho influencia sistemas RFID. A finalidade é analisar esses protocolos e concluir qual protocolo tem melhor desempenho em qual caso e tentar analisar o motivo da queda de rendimento de certos protocolos diante de um ambiente exposto, como por exemplo, o desempenho de um protocolo diante do *Capture – Effect*, o efeito de captura. O efeito de captura é um efeito que ocorre na vida real nos sistemas RFID e que pode alterar completamente o desempenho de um protocolo anti-colisão e conseqüentemente de um sistema RFID na prática. Ele raramente é

estudado na literatura científica o que causa que um protocolo não tenha na prática o mesmo desempenho do estudado numa simulação de laboratório.

1.3.Estrutura

Este trabalho será iniciado com o segundo capítulo mostrando o estado da arte do problema. Apresentando como funciona o sistema RFID e no que consiste uma colisão no sistema RFID e como ela funciona. Aí será apresentado o Capture – Effect, mostrando o que é e como ele pode influenciar negativamente ou positivamente o desempenho um protocolo num ambiente real.

Em seguida, serão mostradas as duas principais categorias de protocolos anti-colisão existentes, os baseados em ALOHA e os baseados em Árvore, aprofundando mais o ALOHA até chegar ao Dynamic Framed-Slotted ALOHA, o objetivo principal deste trabalho. Serão então mostrados os protocolos do tipo *Dynamic Framed-Slotted ALOHA* que serão estudados neste trabalho, o Vogt, o Eom-Lee, o Schoute e o Lower Bound.

No terceiro capítulo falarei sobre como ocorreu a análise de desempenho desses protocolos, desde a validação do simulador até os resultados das simulações e quais conclusões podemos tirar dos resultados.

O trabalho será encerrado com a conclusão dos pontos a serem destacados e com a apresentação das principais contribuições resultantes dele.

2. Conceitos Básicos

2.1. Sistemas RFID

Um sistema RFID é composto por 2 principais componentes, as etiquetas e o leitor[6]. Também fazem parte de sistemas RFID, antenas que normalmente estão integradas ao leitor e às etiquetas, middlewares que funcionam como interfaces de comunicação entre o leitor e o software e o próprio software. Nesta seção, serão apresentados os dois principais componentes e como ocorre a interação entre eles.

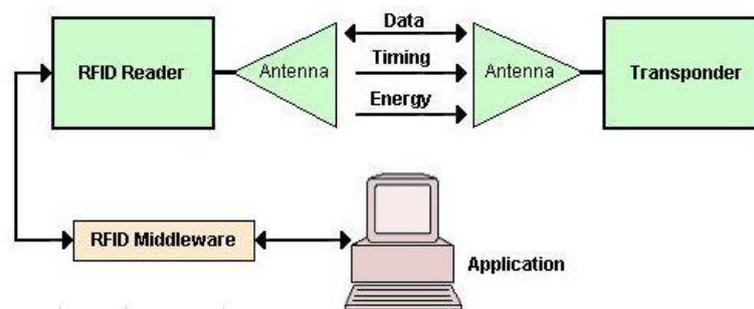


Figura 1: Componentes de um sistema RFID.

2.1.1. Leitores RFID

Os leitores se comunicam com as etiquetas através de sinais de rádio frequência que são enviados pela sua antena. O alcance de um leitor depende da capacidade de transmissão de sinal de sua antena e da potência para a qual está programado a enviar o sinal.

Os leitores são normalmente controlados pela aplicação. Eles têm a capacidade de ler e escrever os dados das etiquetas à distância e de enviar às etiquetas os comandos que controlam as rodadas de leitura.

Entre os comandos que um leitor podem enviar às etiquetas podemos destacar quatro. Seleção/ Anti-Colisão, Escrever, Ler e Ler Não-Selecionados [2].

Seleção/Anti-Colisão (*Anti-collision/Select - ACS*) – Esse comando faz com que todas as etiquetas que estão ao alcance do leitor respondam com seu número de identificação. Esse comando é usado nas rodadas anteriores à rodada de escritura já que a pessoa precisa conhecer o ID de uma etiqueta para poder escrever nela. Esse tipo de comando demora muito mais do que uma rodada de Ler Não-Selecionados.

Ecrever – Esse comando é utilizado em etiquetas nas quais informações podem ser escritas e nas quais pretendemos escrever algo. O ID da etiqueta tem que ser previamente conhecido para se poder escrever informações nela.

Ler – Esse comando faz com que somente etiquetas identificadas sejam lidas. Também tem que conhecer previamente o ID da etiqueta.

Ler Não-Selecionados – Esse lê todas as etiquetas que estão ao alcance do leitor. Esse comando é o mais eficiente para se ler as etiquetas e é simulando esse comando que serão realizadas todas as simulações.

2.1.2. Etiquetas RFID

Etiquetas RFID são objetos que tem capacidade de armazenar e transmitir dados. Elas são compostas por antenas e chips e podem ter ou não baterias. Existem três tipos de etiquetas: passivas, semi-passivas e ativas.

As passivas são etiquetas que não possuem fonte de energia e se comunicam com os leitores através do mecanismo do *backscattering*, onde a etiqueta utiliza o próprio sinal proveniente do leitor para enviar a resposta. Essa técnica funciona como algo semelhante a uma reflexão do sinal. Essas etiquetas são mais baratas e menores e é o modelo de propagação de sinal dessas etiquetas que será considerado nas simulações.

Etiquetas semi-passivas são etiquetas que possuem bateria mas que utilizam essa bateria apenas para energizar seu chip interno. O mecanismo de envio de sinal é o mesmo *backscattering* das passivas. Apesar do mecanismo ser o mesmo o sinal das semi-passivas acaba sendo maior já que ela não necessita de energia para a eletrônica interna e a energia do sinal enviada pelo leitor é utilizada apenas para enviar o sinal de resposta.

Etiquetas ativas são etiquetas que possuem bateria e que utilizam essa bateria em todas as suas funções. Ela não necessita da energia do sinal do leitor e por isso pode enviar sinais ao leitor mesmo sem ser requisitada. Essas etiquetas costumam ter um alcance maior que qualquer outra.

2.2. Colisões em Sistemas RFID

Colisão é o nome dado para quando duas ou mais etiquetas respondem o leitor ao mesmo tempo num processo de leitura de etiquetas RFID. Esse fenômeno acontece quando o sinal de duas ou mais etiquetas colidem, uma causa interferência no sinal da outra fazendo com que o sinal não chegue corretamente ao leitor [2]. Quando isso acontece, ocorre a chamada colisão de dados, o leitor descarta os dados recebidos e aquela rodada de leitura foi desperdiçada [7].

A colisão é um grande problema na leitura de etiquetas RFID, ela causa desperdício de banda, de energia e ainda aumenta o tempo de atraso na identificação das etiquetas[8].

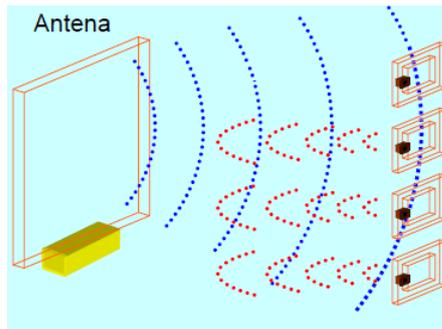


Figura 2: Esquema de leitura de etiquetas com colisão.

Daí vem a necessidade da criação e utilização de protocolos anti-colisão no processo de leitura de etiquetas RFID. Eles são responsáveis por evitar ou pelo menos minimizar as colisões fazendo com que a leitura seja feita de forma mais eficiente.

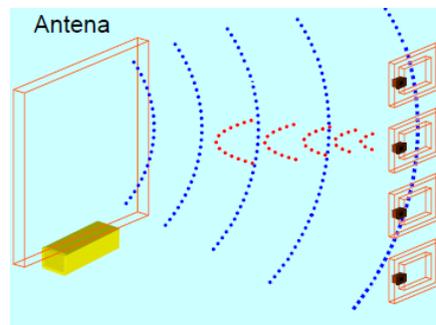


Figura 3: Esquema de leitura de etiquetas sem colisão.

2.3. Protocolos Anti-Colisão

Protocolos anti-colisão são os que têm o intuito de evitar que diversas etiquetas se comuniquem ao mesmo tempo no mesmo canal evitando que o leitor não consiga identificá-las.

Os principais tipos de protocolos para sistemas RFID são os protocolos baseados em árvore e os baseados em ALOHA. Os dois tipos são protocolos TDMA (*Time-Division Multiple Access*), ou seja, eles funcionam dividindo seu canal de comunicação pelo tempo. Cada protocolo, de acordo com seu funcionamento, reserva certa quantidade de tempo para que a comunicação seja realizada.

Os protocolos baseados em ALOHA são probabilísticos e têm como objetivo diminuir a probabilidade de colisões enquanto que os baseados em árvores são determinísticos e tentam garantir que encontrarão a situação na qual não vai haver colisão [9].

Abaixo tem um organograma com os principais protocolos anti-colisão. Os baseados em árvore não são objetos de estudo desse trabalho e por isso não falaremos muito sobre eles. Já para o ALOHA, será apresentada a evolução do ALOHA puro até o *Dynamic Frame-Slotted* ALOHA passando pelo *Slotted* ALOHA e pelo *Frame-Slotted* ALOHA.

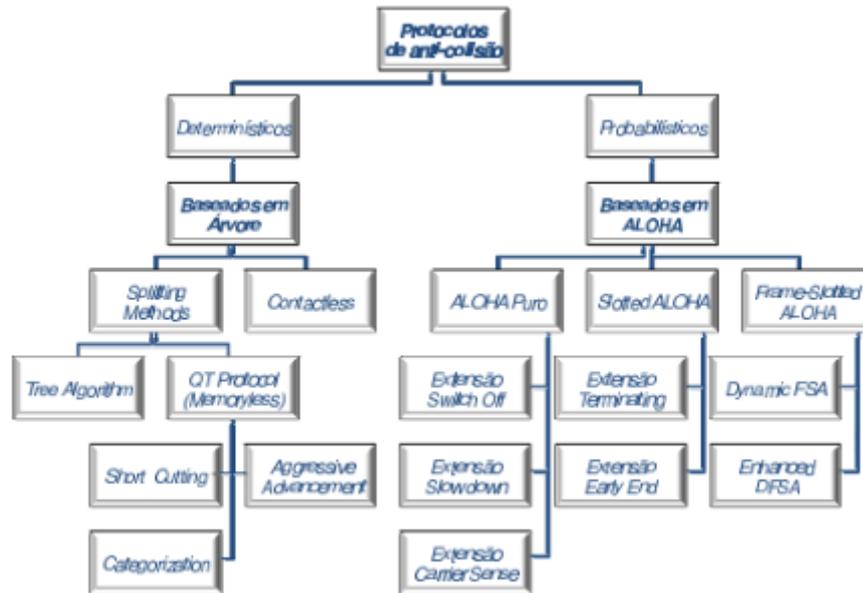


Figura 4: Organograma dos protocolos anti-colisão.

2.3.1. Protocolos Baseados em Árvore

Protocolos baseados em árvore são protocolos que são “Reader-Driven”, ou seja, o protocolo é feito pensando e baseando-se no leitor. Esses protocolos normalmente tem como objetivo dividir as etiquetas em grupos até que consiga formar grupos unitários. Quando o leitor isola uma etiqueta, ele então realiza a leitura[9].

Um exemplo de protocolo baseado em árvore é o QT, Query Tree Protocol, que envia requisições para etiquetas cujo id comece com um determinado prefixo e, se mais de uma etiqueta responder, ele saberá que precisará especificar mais o prefixo [10]. Um exemplo típico de funcionamento do QT seria um leitor que requisita etiquetas cujo id comece com qualquer prefixo e obtém uma colisão. Ele então em seguida requisita etiquetas que comecem em “1” e depois em “0”. Ao obter novas colisões ele teria que requisitar etiquetas com prefixos “00”, “01”, “10” e “11”. O leitor só para de decompor um determinado prefixo quando não obtém nenhuma resposta, que é o caso do canal vazio, ou quando obtém a resposta de apenas uma etiqueta, sucesso.

Abaixo temos a representação do funcionamento do QT para um caso onde temos quatro etiquetas que só podem ser separadas a partir de um prefixo com três dígitos [9]. Repare que todas as folhas da árvore tem que ser necessariamente slots vazios ou de sucesso e que quando um desses dois casos acontecem a decomposição desse prefixo para.

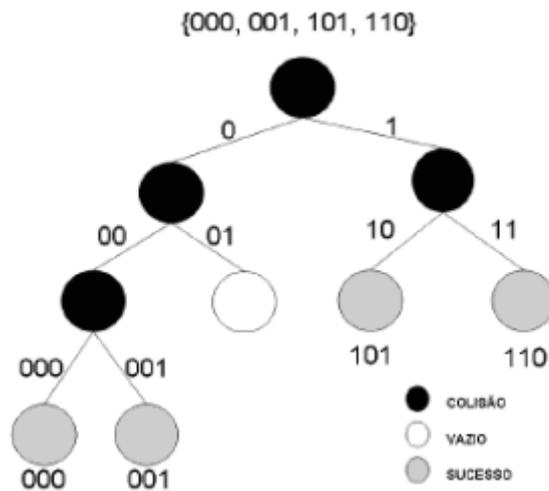


Figura 5: Arvore que simula o funcionamento do QT.

2.3.2. Protocolos ALOHA

Os Protocolos ALOHA se baseiam na idéia de que uma colisão deve ser tratada quando ocorre, mas não há nada determinístico para impedir que ela ocorra. Ela funciona com as etiquetas enviando seu ID ao leitor simplesmente quando estão prontas. Quando ocorre de duas ou mais etiquetas enviarem a informação ao mesmo tempo e a colisão ocorre, as etiquetas param de transmitir por aquele momento e re-transmitem depois de um determinado tempo aleatório.

As colisões nesse tipo de protocolo podem ser parciais, quando uma etiqueta começa a transmitir antes da outra, ou totais, quando as duas começam a transmitir ao mesmo tempo. Considera-se que as parciais são mais prejudiciais ao sistema já que elas fazem com que uma parte dos dados já enviados e recebidos sejam descartados além de ter que esperar a etiqueta que começou depois terminar para que o sistema possa começar novamente uma comunicação sem colisão[11].

Na figura abaixo vemos como se desenrola o ALOHA.

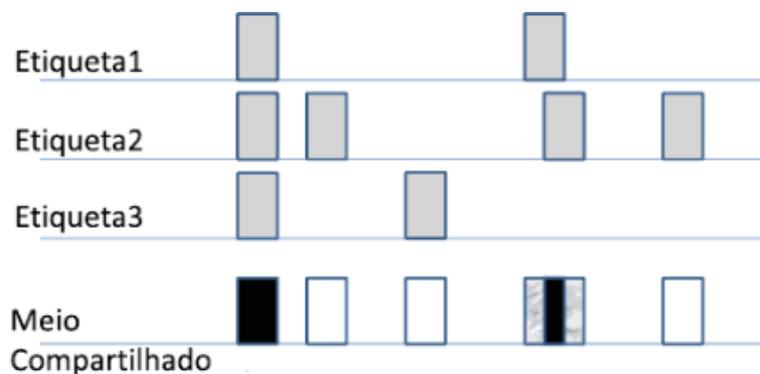


Figura 6: Esquema que simula o funcionamento do ALOHA.

2.3.2.1. Slotted Aloha

O protocolo *slotted aloha* é uma extensão do ALOHA puro e apesar de seguir o mesmo princípio ele introduz algo novo no sistema que são os *slots*. Slots nada mais são do que intervalos de tempo nos quais a etiqueta tem que enviar seus dados. Esse mecanismo evita as colisões parciais já que as etiquetas sempre tem que começar a enviar no começo de um *slot* e sempre terminam de enviar seus dados dentro desse mesmo *slot* [11].

Para poder realizar o *slotted ALOHA* é sempre necessário sincronizar as etiquetas no começo da rodada de leitura para que todas estejam cientes de quando começa e termina um slot.

Como dito anteriormente aqui é seguido o mesmo princípio do ALOHA, então para enviar seus dados, as etiquetas escolherão aleatoriamente um *slot* e no caso de haver colisão deverão tentar em outros *slots*, também escolhidos aleatoriamente, até que todas as etiquetas tenham enviado suas informações.

Na figura abaixo vemos como funciona o Slotted ALOHA.

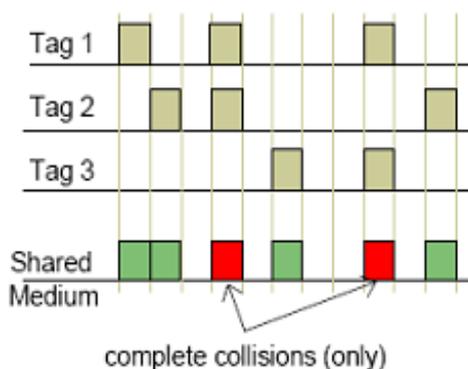


Figura 7: Esquema que simula o funcionamento do Slotted ALOHA.

2.3.2.2. Frame-Slotted Aloha

O *frame-slotted ALOHA* é um protocolo que tem como base o Slotted-Aloha mas que introduz um conceito novo, o conceito do *frame*. O *frame* nada mais é do que um conjunto de N *slots*. A introdução do *frame* é para obrigar as etiquetas a escolherem apenas um *slot* por *Frame*, evitando assim que uma mesma etiqueta cause diversas colisões consecutivas[11].

Para executar o *frame-slotted ALOHA* as etiquetas precisam conhecer o tamanho do *frame*. A grande vantagem da introdução do *frame* é a garantia de que a frequência de envio de uma etiqueta nunca será maior que uma por *frame*.

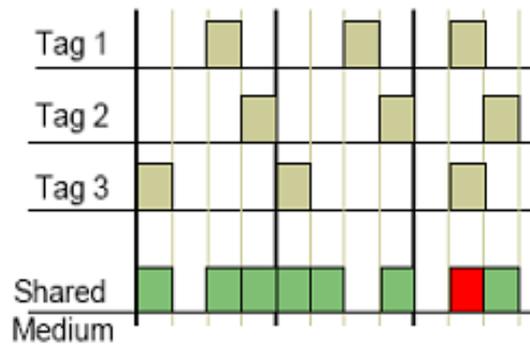


Figura 8: Esquema que simula o funcionamento do Frame-Slotted ALOHA.

A grande desvantagem do *Frame – Slotted ALOHA* é que se tivermos um número de etiquetas muito maior do que o tamanho do *frame* a *performance* do algoritmo será quase nula. Para evitar esse problema é que existe o *Dynamic Frame-Slotted ALOHA*, que será mostrado logo a seguir.

2.3.2.3. *Dynamic Frame-Slotted Aloha*

O *Dynamic Frame-Slotted Aloha* é um protocolo do tipo *Frame-Slotted Aloha* cujo *frame* varia de uma rodada de leitura para a outra. Isso é feito para tentar fazer com que sejam gastos a menor quantidade de tempo e de slots possíveis numa rodada de leitura.

Para se gastar a menor quantidade possíveis de *slots* é necessário tentar sempre aproximar o tamanho do *frame* à quantidade de *Tags* [2]. É justamente isso que os protocolos desse tipo tentam fazer. O caso ideal seria saber quantas etiquetas devem ser lidas e colocar como o tamanho do frame, mas num sistema real o leitor nunca sabe quantas etiquetas estão presentes no seu alcance antes de entrar em contato com elas.

Para realizar então a estimativa de etiquetas, os algoritmos de cálculo do tamanho do frame recebem um vetor com as três variáveis de uma rodada de leitura e tentam calcular o tamanho ideal para o próximo *frame* [12].

As três variáveis usadas no protocolos são: quantidade de *slots* vazios, quantidade *slots* em colisão e quantidade de *slots* de sucesso. Repare que com esse vetor temos também o tamanho do frame atual já que ela é composta pela soma dos 3 valores.

Abaixo está descrito como deve funcionar na teoria um protocolo do tipo DFSA.

1. Realizar a rodada de leitura com o tamanho do Frame inicial
 2. Enquanto Todas as etiquetas não tiverem sido lidas
 3. Calcule o Novo Tamanho do Frame (colisões, sucessos, vazios)
- Retorna Novo Tamanho do Frame

4. Realize nova rodada de leitura com o o novo tamanho do frame
– Retorna (colisões, sucessos, vazios)
5. Terminar Enquanto

Figura 9: Fluxo do funcionamento do Dynamic Frame-Slotted ALOHA.

O que diferencia um protocolo DFSA para o outro é a função de cálculo do tamanho do frame que pode ser visto no algoritmo acima na linha 3. Na próxima seção estudaremos quatro diferentes protocolos DFSA e veremos como cada um deles fazem para calcular o tamanho do frame.

2.4. Protocolos DFSA Estudados

Foram escolhidos quatro protocolos para serem analisados nesse artigo, todos do tipo Dynamic Frame-Slotted Aloha. Os quatro protocolos analisados foram o Lower Bound, o Schoute, o Vogt e o Eom-Lee. Cada um deles será apresentado a seguir.

2.4.1. Lower Bound

O Lower Bound é o protocolo mais simples do tipo Dynamic Frame-Slotted Aloha. Ele tem o nome de Lower Bound porque é o que faz sempre a menor estimativa possível para o número total de tags, ele é o limiar inferior das estimativas.

O Lower Bound é o limiar inferior porque parte do princípio de que cada colisão ocorre por apenas 2 etiquetas. Partindo dessa premissa ele estima a quantidade total de etiquetas no sistema como sendo o número de etiquetas lidas com sucesso somado ao número de colisões ocorridas multiplicadas por dois.

$$\text{Numero Total de Etiquetas} = \text{Sucessos} + \text{Colisões} * 2$$

Como as etiquetas de sucesso já foram lidas, o tamanho do próximo frame deve ser o número de etiquetas que restam, também conhecido como backlog. O tamanho do próximo frame será igual ao número de colisões multiplicadas por dois.

$$\text{Backlog} = \text{Colisões} * 2$$

2.4.2. Schoute

Schoute parte do mesmo princípio do protocolo Lower Bound e define que para cada slot em colisão há um número definido de etiquetas que ocupam esse slot [13]. Então no protocolo Schoute também temos :

Numero Total de Etiquetas = Sucessos + Colisões * Constante

E

Backlog = Colisões * Constante

A principal e única diferença de Schoute para o Lower Bound é que a definição dessas constantes não partem apenas de pressupostos teóricos mas também de cálculos estatísticos. Usando a distribuição de Poisson ele chega à conclusão que a probabilidade de K dispositivos escolherem um determinado slot é :

$$\Pr \{k \text{ mobiles choose slot } i\} \triangleq p_k = e^{-1} \cdot \frac{1}{k!}.$$

Considerando que um slot em colisão tem no mínimo 2 etiquetas e que pode ter no máximo infinitas etiquetas, temos que a fórmula para calcular o numero médio de etiquetas em cada slot que está em colisão é:

$$\sum_{k=2}^{\infty} k \cdot p_k = \frac{\sum_{k=2}^{\infty} e^{-1} \frac{1}{(k-1)!}}{1 - 2 \cdot e^{-1}} = \frac{e-1}{e-2} \approx 2.39.$$

Após resolver a fórmula, Schoute chegou ao valor de 2,39 . Por isso as fórmulas do protocolo de Schoute são:

Numero Total de Etiquetas = Sucessos + Colisões * 2.39

E

Backlog = Colisões * 2.39

2.4.3. Vogt

O protocolo de Vogt tem aproximadamente oito anos e tem uma abordagem bastante diferente para calcular o número estimado de etiquetas. Ele se baseia na desigualdade de Chebyshev onde calcula a diferença entre o vetor real de slots com colisão, sucesso e vazios entre com um vetor de estimativa entre esses mesmos 3 parâmetros[2].

$$\begin{pmatrix} a_0^{N,t} \\ a_1^{N,t} \\ a_k^{N,t} \end{pmatrix} - \begin{pmatrix} c_0 \\ c_1 \\ c_k \end{pmatrix}$$

A estimativa dos slots em colisão, sucesso e vazios se baseia no número total de etiquetas estimadas t e no tamanho do frame N naquele momento. Considerando esse dois fatores temos a fórmula probabilística que calcula a probabilidade de r etiquetas ocuparem um mesmo slot.

$$a_r^{N,t} = N \times \binom{t}{r} \left(\frac{1}{N}\right)^r \left(1 - \frac{1}{N}\right)^{t-r}$$

Considerando essa fórmula probabilística temos as fórmulas para estimativa dos 3 elementos do vetor de estimativa logo abaixo. a_0 sendo a estimativa de slots vazios, a_1 sendo a estimativa de slots com sucesso e a_k sendo a estimativa de slots em colisão.

$$a_0^{N,t} = N \times \left(1 - \frac{1}{N}\right)^t$$

$$a_1^{N,t} = t \times \left(1 - \frac{1}{N}\right)^{t-1}$$

$$a_k^{N,t} = N - a_0^{N,t} - a_1^{N,t}$$

O valor estimado do total de etiquetas vai ser o número estimado de etiquetas para o qual a diferença entre os vetores for o mínimo possível.

$$\text{Número Total de Etiquetas Estimadas} = \min_t \left| \begin{pmatrix} a_0^{N,t} \\ a_1^{N,t} \\ a_k^{N,t} \end{pmatrix} - \begin{pmatrix} c_0 \\ c_1 \\ c_k \end{pmatrix} \right|$$

Para se chegar ao mínimo basta fazer um simples algoritmo que começa estimando o total de etiquetas como sendo igual a zero e incrementamos até chegar ao mínimo da função.

E então calculamos o backlog, tamanho do próximo frame como sendo o número restante de etiquetas.

$$\text{Backlog} = \text{Número Total de Etiquetas Estimadas} - \text{Sucessos}$$

Esse protocolo é mais indicado para sistemas com uma alta densidade de etiquetas [14] e é o protocolo que exige o maior poder computacional dentre os estudados nesse trabalho [15].

2.4.4. Eom-Lee

O protocolo Eom-Lee é o mais novo dentre todos os estudados por esse trabalho e a maior motivação para o estudo realizado. Ele foi divulgado há menos de um ano em [15] e apresentou resultados de performance superiores a todos os outros protocolos conhecidos em seu artigo.

Eom-Lee parte do mesmo princípio que os protocolos mais antigos de que o backlog tem que ser :

$$\text{Backlog} = \text{Colisões} * \text{Constante}$$

Nesse protocolo essa constante essa que é chamada de γ . Porém há uma grande diferença nessa constante de Eom-Lee, ele considera que ela deve ser calculada a cada rodada já que o número médio de etiquetas por colisão varia em cada rodada de leitura.

$$\text{Backlog} = \text{Colisões} * \gamma$$

O protocolo eom-lee também introduz outra constante que é β . Essa constante é o fator ideal que se deve multiplicar o número de etiquetas para se chegar ao tamanho do Frame.

$$\text{Tamanho do Frame} = \text{Número de Etiquetas} * \beta$$

Para encontrar o valor de γ e β tem-se que tecer algumas considerações. Primeiramente deve-se considerar a probabilidade de que r dentre n etiquetas ocupe um determinado slot como sendo:

$$P_r = \binom{n}{r} \left(\frac{1}{L}\right)^r \left(1 - \frac{1}{L}\right)^{n-r}.$$

A partir da premissa de que L seja um número suficientemente grande já podemos calcular P_0 , a probabilidade de slots vazios.

$$P_0 = \left(1 - \frac{1}{L}\right)^n \approx e^{-\frac{n}{L}}.$$

E pode-se também calcular P_1 , a probabilidade de slots de sucesso.

$$P_1 = n \frac{1}{L} \left(1 - \frac{1}{L}\right)^{n-1} \approx n \frac{1}{L} \left(\frac{L}{L-1}\right) e^{-\frac{n}{L}}.$$

Deduzindo a equação ,

Colisões * γ = Total de Etiquetas – Sucesso

A partir dos valores de P0, P1, γ e β chega-se à conclusão de que :

$$\gamma = \frac{1 - e^{-\frac{1}{\beta}}}{\beta \left(1 - \left(1 + \frac{1}{\beta}\right) e^{-\frac{1}{\beta}}\right)}.$$

$$backlog = \gamma \cdot s_c = \frac{1 - e^{-\frac{1}{\beta}}}{\beta \left(1 - \left(1 + \frac{1}{\beta}\right) e^{-\frac{1}{\beta}}\right)} \cdot s_c.$$

Se γ e β forem considerado como valores iterativos e aplicarmos regra de l'hôpital for aplicada em γ para $\gamma \rightarrow 1$, temos :

$$\gamma_1 = \lim_{\beta_1 \rightarrow \infty} \frac{1 - e^{-\frac{1}{\beta_1}}}{\beta_1 \left(1 - \left(1 + \frac{1}{\beta_1}\right) e^{-\frac{1}{\beta_1}}\right)} = 2.$$

A partir de γ_1 é possível deduzir β_2 ,

$$\beta_2 = \frac{L}{\gamma_1 \cdot s_c + s_s} < \beta_1,$$

E a partir de β_2 é possível deduzir γ_2 ,

$$\gamma_2 = \frac{1 - e^{-\frac{1}{\beta_2}}}{\beta_2 \left(1 - \left(1 + \frac{1}{\beta_2}\right) e^{-\frac{1}{\beta_2}}\right)}.$$

Agora foi deduzido um algoritmo iterativo para tentar encontrar o valor ideal para as constantes :

$$\beta_k = \frac{L}{\gamma_{k-1} \cdot s_c + s_s},$$

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

Como a sequência β_k converge para um limite, pode se considerar os valores $\gamma_1 = 2$ e $B_1 = \infty$ para iniciar a iteração e descobrir o valor limite de γ_k na iteração k . A condição de parada para considerar que se chegou no limite é que se tenha uma variação muito pequena de uma rodada para outra, ou seja de γ_{k-1} para γ_k . Esse valor da condição de parada é o *threshold*.

$$|\gamma_{k-1} - \gamma_k| < \epsilon_{threshold}$$

O valor indicado por Eom-Lee para o *threshold* é 0.001 e foi exatamente esse o que foi utilizado neste trabalho.

Quando finalmente o valor de γ é encontrado, o cálculo do deve ser feito backlog através da equação :

$$\text{Backlog} = \text{Colisões} * \gamma.$$

2.5. Capture Effect

Capture Effect é um termo em inglês que significa efeito de captura. Esse termo vem da área de telecomunicações e é utilizado para quando dois sinais na mesma faixa de frequências são recebidos e o que possuir maior nível de recepção será aceito e demodulado enquanto o outro será descartado [16].

Em sistemas RFID ele se refere ao fenômeno que ocorre quando mais de uma etiqueta responde o leitor ao mesmo tempo, mas mesmo assim o leitor captura corretamente o sinal de uma das etiquetas [2]. Ou seja, o *capture effect* se refere a quando uma colisão deveria ocorrer na teoria, mas na prática não ocorre.

Para que o *capture effect* ocorra é necessário que o sinal proveniente de uma das etiquetas seja consideravelmente mais forte do que o vindo das outras etiquetas [3]. Quando isso ocorre a etiqueta que envia o sinal mais forte é corretamente lida pelo leitor.

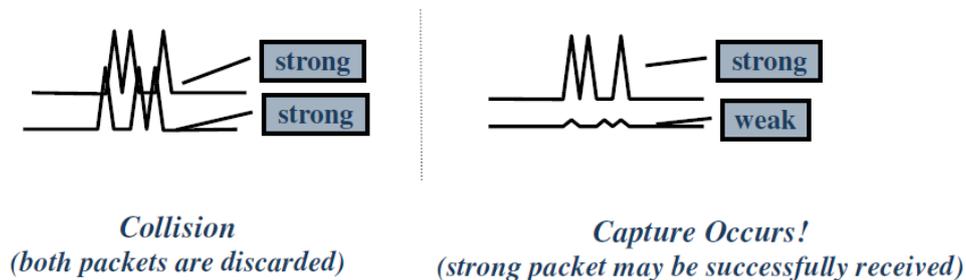


Figura 10: Diferença de sinais numa colisão e na ocorrência do capture effect.

2.5.1. Calculando o Capture Effect

Há diversas maneira de simular e verificar a ocorrência do *capture effect* ou não. Em [3] são apresentados três modelos para essa verificação, dentre os quais o modelo SIR. O modelo SIR é o que foi usado nas simulações de [3] e que será também usado nesse trabalho.

O Modelos SIR, *signal to interference ratio*, cria um coeficiente entre o sinal de rádio mais forte, P_i , que normalmente vai ser o da etiqueta mais próxima e o somatório dos sinais de todas as outras etiquetas que transmitem ao mesmo tempo. Se esse coeficiente for maior que o *capture ratio*, z_0 , então a captura da etiqueta de sinal mais potente acontece.

$$P_{cap}(N) = N \cdot P\left(\frac{P_i}{\sum_{j=1, j \neq i}^{N-1} P_j} > z_0\right)$$

O modelo para o cálculo da potência do sinal recebido pelo leitor também foi extraído de [3] e é o que se encontra logo abaixo.

$$P_i, \text{ or } P_j = P_T \cdot \left(\frac{\lambda}{4\pi d_0}\right)^2 d^{-n} R^2 g e \cdot \eta \cdot g \left(\frac{\lambda}{4\pi d_0}\right)^2 d^{-n} R^2$$

Pt	Potência de Transmissão
λ	Frequência
Π	Pi
d0	Distância de Referência
R	Fator Ricean
g	Ganho da Antena
z0	Quociente de Captura
e	Eficiência de Polarização
n	Expoente de atenuação
η	Quociente de Retroespalhamento

Figura 11: Nome das variáveis presentes na fórmula de cálculo da potência de sinal recebido.

3. Análise

O simulador foi construído em C e teve como objetivo simular um protocolo DFSA com ou sem captura. O método de cálculo do tamanho do frame do DFSA é ajustável e pode ser Lower Bound, Schoute, Vogt ou Eom-Lee.

Os resultados são extraídos tirando-se a média de 1000 simulações.

3.1. Validação do Simulador

3.1.1. Informações Avaliadas

Nesse trabalho, seguindo a tendência dos principais artigos, foi decidido fazer dois tipos de extrações, a da quantidade de tags para um determinado número de slots e a avaliação da performance dos protocolos considerando a *performance* como sendo uma relação entre a quantidade de tags e o número de slots utilizados para lê-las.

A grande vantagem da análise de quantidade de tags para um número determinado de slots que pode ser encontrada em [15][17] é a possibilidade de ver um número facilmente compreensível que mostra realmente o que é necessário para realizar uma leitura de etiquetas. Essa análise consiste simplesmente em realizar leituras para descobrir quantos *slots* são necessários para ler um determinado número de *tags*.

A outra informação avaliada, a da *performance*, pode ser encontrada em [14][3] e permite que a pessoa veja a taxa de sucessos daquele protocolo. Esse método também facilita um cálculo de eficiência média de cada protocolo. O cálculo da *performance* consiste em :

Número de Etiquetas/*Slots* necessários

A média é simplesmente a média de cada um dos casos analisados.

3.1.2. Validação do DFSA

O artigo utilizado para comparar e validar os resultados da simulação foi o [15]. Nele realiza-se uma simulação de protocolos DFSA sem *capture effect* com um *frame* inicial de 64 e com o número de etiquetas variando de 100 a 1000.

Nessa simulação, o objetivo era ver quantos *slots* eram necessários para identificar um numero x de etiquetas em cada protocolo.

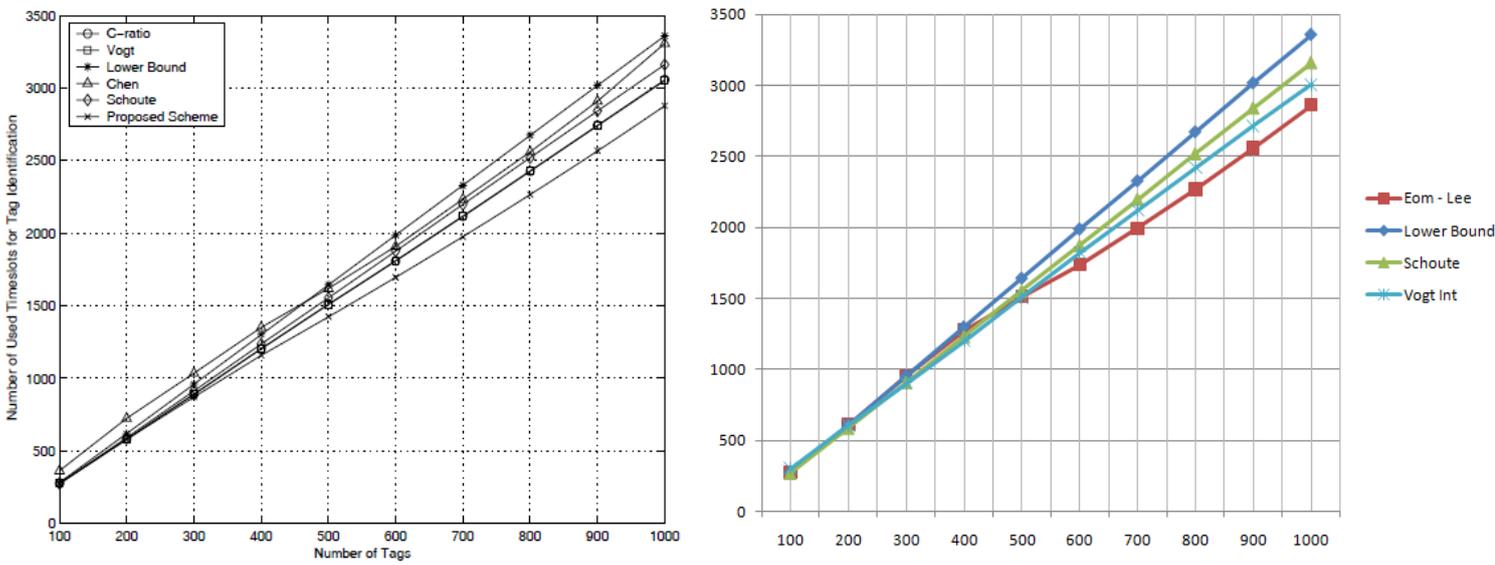


Figura 12: Comparação do DFSA entre o grafico de [15], da esquerda, e o tirado pelo simulador deste trabalho, da direita.

3.1.3. Validação do *Capture Effect*

O artigo utilizado para comparar e validar os resultados da simulação capture effect foi o [3]. O objetivo DFSA é simular o *capture effect* num modelo DFSA já validado. Foi utilizado um *frame* inicial de 128 e o número de etiquetas variou de 100 a 1000.

Nessa simulação o objetivo era ver a diferença de eficiência com e sem *capture effect* para cada um dos 3 protocolos presentes no artigo. O Lower Bound, o Schoute e o Vogt.

Os valores utilizados na simulação são os mesmos que [3].

P_t	Potência de Transmissão	20 DB
λ	Frequência	866 MHz
Π	Pi	3.14
d_0	Distância de Referência	1 m
R	Fator Ricean	6 DB
g	Ganho da Antena	1 DB
z_0	Quociente de Captura	6 DB
e	Eficiência de Polarização	0.5
n	Expoente de atenuação	2

Figura 13: Valor das variáveis presentes na formula de calculo da potência de sinal recebido.

Em [3] foi feito um número menor de simulações e foi utilizado um intervalo de confiança de 95%. Neste trabalho optou-se por um grande número de simulações com a extração da média de todos os resultados. A tendência com um número maior de simulações é obter um comportamento mais uniforme e fiel do protocolo. Esses resultados serão observados através de gráficos. Por esses motivos citados acima as tabelas não serão exatamente iguais, mas mesmo assim devem expor resultados similares e com valores aproximados, e é exatamente isso que elas fazem.

3.1.3.1. Lower Bound

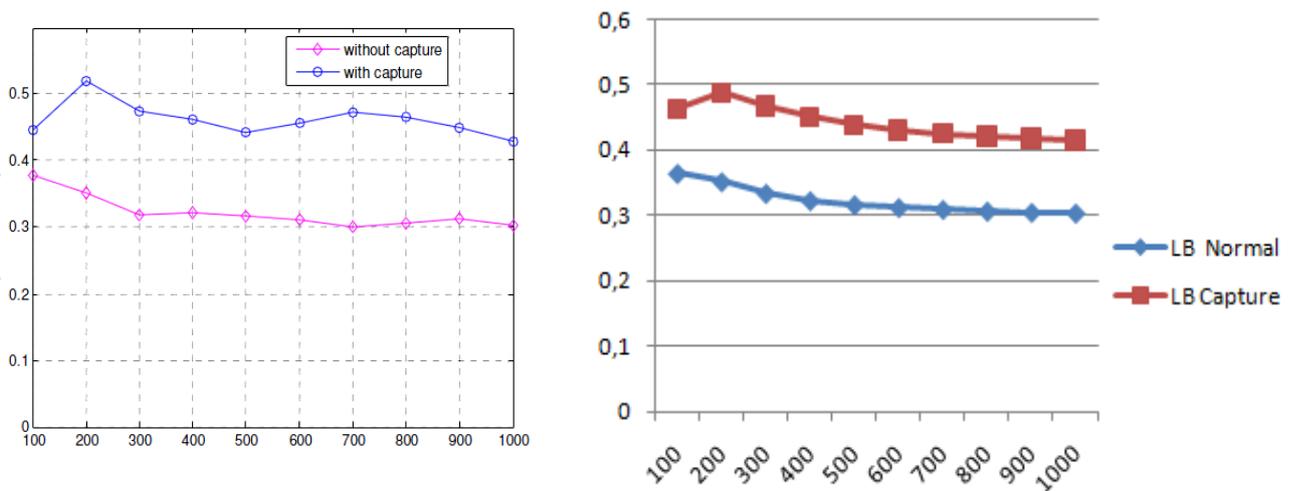


Figura 14: Comparação do capture effect para o Lower Bound entre o grafico de [3], da esquerda, e o tirado pelo simulador deste trabalho, da direita.

3.1.3.2. Schoute

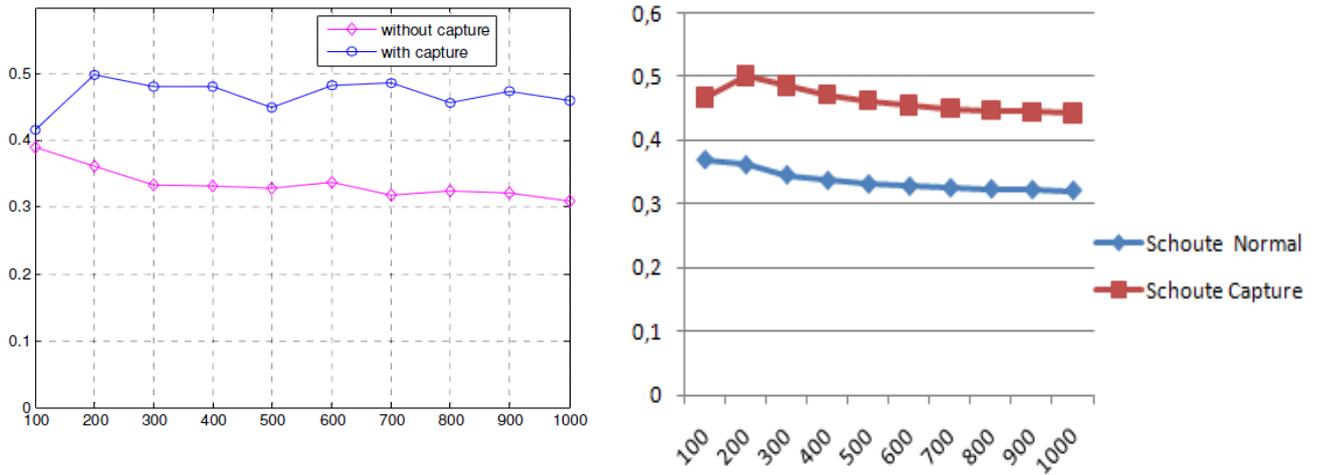


Figura 15: Comparação do capture effect para o Schoute entre o grafico de [3], da esquerda, e o tirado pelo simulador deste trabalho, da direita.

3.1.3.3. Vogt

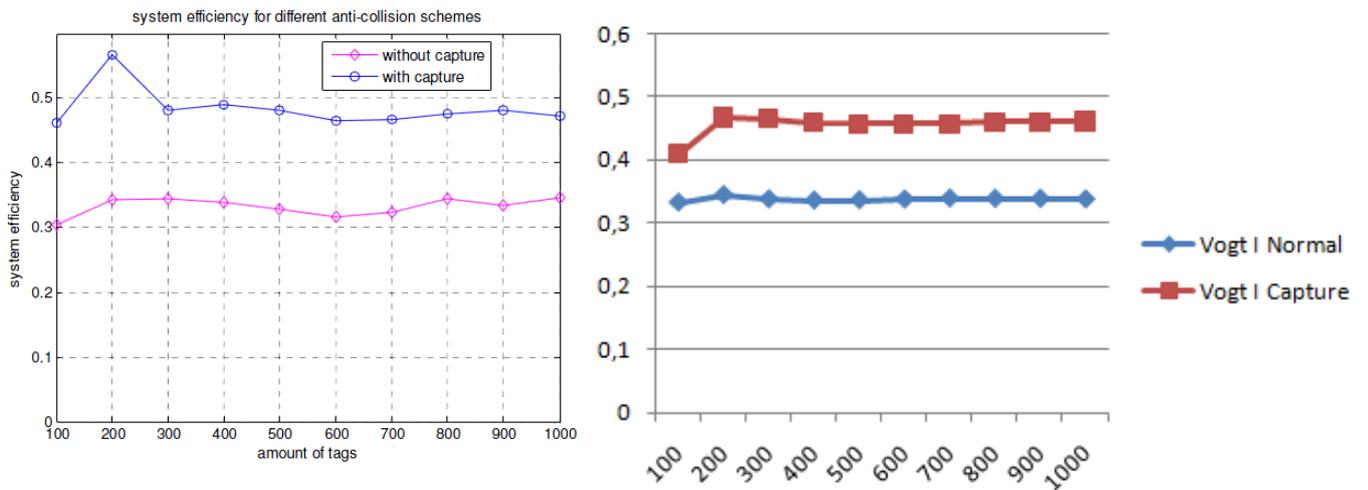


Figura 16: Comparação do capture effect para o Vogt entre o grafico de [3], da esquerda, e o tirado pelo simulador deste trabalho, da direita.

3.2. Simulações

O objetivo das simulações era avaliar o comportamento do algoritmo Eom-Lee em ambientes diferentes do proposto em [15] e compará-lo com os resultados de protocolos já

conhecidos e consolidados. Para isso, foram feitas simulações com variações do *frame* inicial, variações de etiquetas e com ou sem *capture effect*.

A variação do *frame* inicial serve para avaliar o comportamento do protocolo com relação a esse parâmetro que pode variar de uma implementação para a outra. A variação de etiquetas serve para avaliar como o protocolo se comporta em diferentes ambientes desde o reconhecimento de uma feira de supermercado com 100 itens ou no reconhecimento de uma caixa numa cadeia de produção com 5000 canetas.

A utilização do *capture effect* é para avaliar se o protocolo é bom de verdade na prática ou se ele é bom apenas na teoria e deve ser melhorado para que se torne bom em situações reais.

Parâmetro	Minimo - Maximo	Variação
Frame Inicial	32-256	x2
Etiquetas	100-5000	100
Distância das Etiquetas ao Leitor	0.3 - 1.0	Aleatorio

Todos os gráficos extraídos das simulações se encontram nos Anexos A, B e C. A seguir serão apresentados alguns gráficos que foram considerados mais importantes, seja porque sintetizam o experimento como um todo ou porque mostram algum fato novo e relevante.

3.2.1. Avaliação da Performance dos Protocolos sem *Capture Effect*

As primeiras simulações foram feitas envolvendo um ambiente sem *capture effect*. Todos os gráficos relacionados às simulações desta seção podem ser encontrados no anexo A.

No ambiente sem *capture effect*, ou seja mais distante de uma simulação de um ambiente real, dependendo do tamanho do *frame* e da quantidade de *slots* observou-se diversas alternâncias sobre qual era o protocolo de melhor desempenho entre o Eom-Lee e o Vogt. Também foram observadas alternâncias entre o Eom-Lee e o Schoute pela segunda colocação. Porém, analisando a eficiência média para cada experimento com um *frame* inicial diferente tivemos sempre o Vogt na primeira posição, contrariando assim os resultados divulgados em [15].

Em [15] seus cálculos resultam em uma melhor eficiência do Eom-Lee porque ele considera apenas o caso específico de um *frame* inicial com tamanho 64 e indo de 100 a 1000 etiquetas (ver fig. 12). Se for observado cuidadosamente o próprio caso que ele escolhe em seu artigo e aumentar a quantidade de etiquetas observa-se que esse protocolo é em pouco tempo superado pelo Vogt.

Aumentando o experimento para 5000 etiquetas, pode-se ver que pouco depois das 1000 etiquetas, onde [15] para, o Vogt se aproxima do Eom-Lee e a partir de 1500 etiquetas é sempre melhor que Eom-Lee.

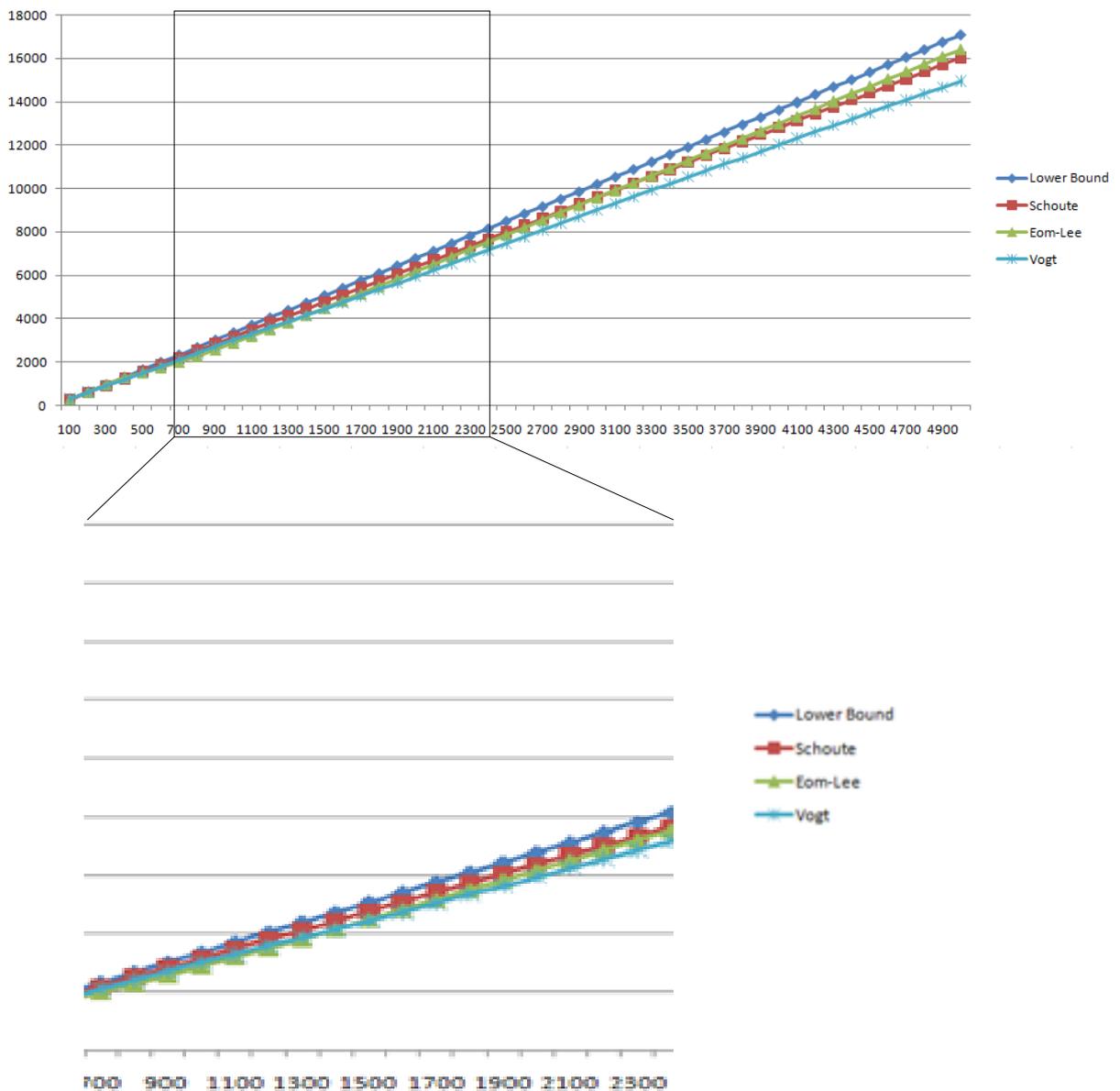


Figura 17: Simulação do DFSA sem capture effect com frame inicial 64 e um Zoom da area que vai de 700 a 2300 etiquetas.

Nesse experimento com *frame* inicial de 64, Vogt apresentou uma eficiência média de aproximadamente 33,35% enquanto Eom-Lee apresentou uma eficiência média de 32,10%, Schoute obteve 31,57% e o Lower Bound 29,71%.

Em todos os quatro experimentos sem *capture effect* a ordem de eficiência dos protocolos foi essa acima. O experimento para o qual o Eom-Lee teve uma eficiência mais próxima ao Vogt foi para a *frame* inicial sendo 256, onde o protocolo Eom-Lee foi superior ao Vogt de 100 a 300 etiquetas e de 2500 a 5000.

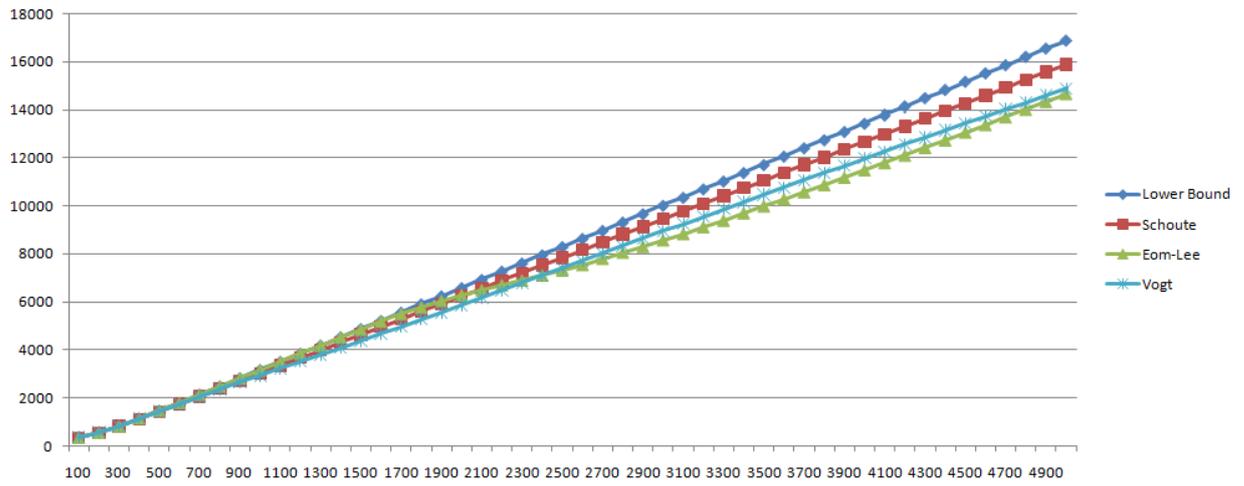


Figura 18: Simulação do DFSA sem capture effect com frame inicial 256.

Nesse experimento com *frame* inicial de 256 o Vogt apresentou uma eficiência média de aproximadamente 33,78% enquanto Eom-Lee apresentou uma eficiência média de 33,58%, Schoute obteve 32,32% e o Lower Bound 30,71%.

Então, pelas quatro simulações pudemos ver realmente que o Vogt sempre teve a melhor *performance* enquanto, Eom-Lee sempre foi o segundo melhor seguido pelo Schoute. O Lower Bound, como já era esperado, sempre apresentou os piores resultados.

1°	Vogt	33,5761877%
2°	Eom-Lee	32,6346782%
3°	Schoute	31,7913722%
4°	Lower Bound	30,0132043%

Figura 19 : Eficiência Média dos Protocolos juntando as quatro simulações sem *capture effect*.

3.2.2. Avaliação do Comportamento dos Protocolos Com *Capture Effect*

A segunda bateria de simulações envolveu um ambiente com *capture effect*. Esse ambiente se aproxima mais de um ambiente real [2] e pode alterar substancialmente os resultados de *performance* de um protocolo [3]. Todos os gráficos relacionados a esta seção podem ser encontrados no anexo B.

Nessa simulação, diferentemente das simulações sem *capture effect*, não foram sempre os mesmos protocolos ocupando as mesmas posições no *ranking* de *performance* para cada *frame* inicial. Apesar de Vogt ser sempre o protocolo mais performante e o Lower Bound ser sempre o menos, houve uma alternância no segundo posto entre o Eom-Lee e o Schoute.

O Eom-Lee foi o segundo melhor para os *frames* iniciais de 64 e 128 com o Schoute em terceiro, enquanto o Schoute foi o segundo melhor para os *frames* iniciais de 32 e 256 mas com uma diferença mínima de 0,1% nos dois casos.

Com o *frame* inicial em 128 observou-se a maior diferença de *performance* entre o protocolo Schoute, que obteve uma eficiência de 43,81%, e o protocolo Eom-Lee que conseguiu capturar etiquetas com sucesso em 45,15% dos seus *slots*. O Vogt, como em todas as simulações até aqui, foi o melhor com uma taxa de sucesso de 45,86%. O Lower Bound apresentou uma taxa de 41% de sucessos.

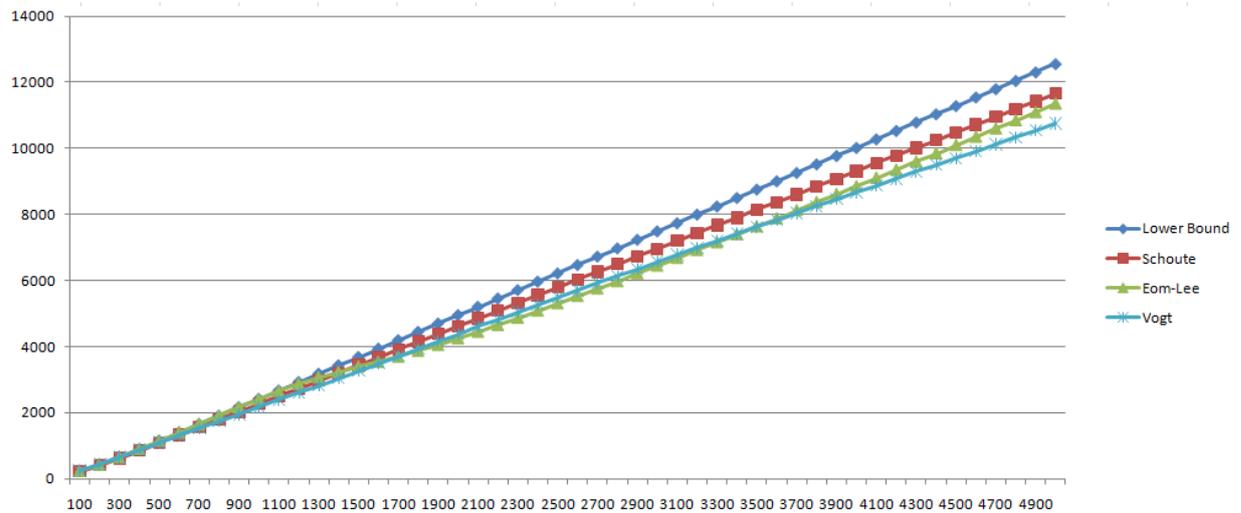


Figura 20 : Simulação do DFSA com capture effect com frame inicial 128.

Analisando o caso do *frame* inicial sendo 32 no qual o Schoute foi superior ao Eom-Lee em *performance* média vemos que a diferença não é muito grande e que para diversos números de etiquetas, o Eom-Lee foi superior.

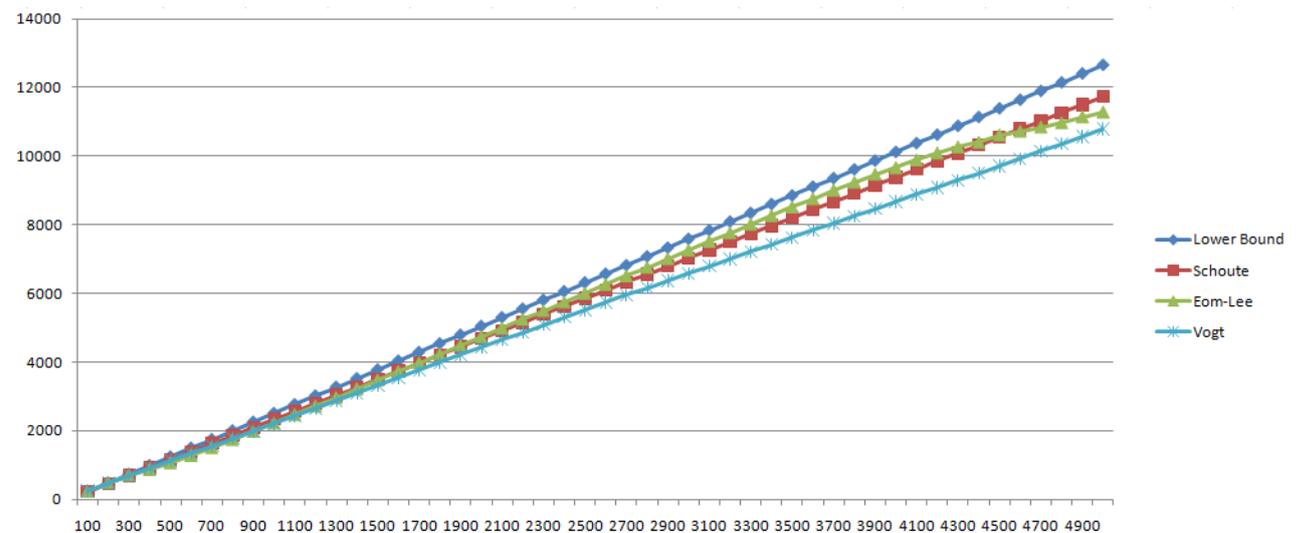


Figura 21 : Simulação do DFSA comm capture effect com frame inicial 32.

Porém ao se tirar a *performance* média, percebemos que o Schoute leva uma leve vantagem de 42,92% de eficiência contra 42,83% do Eom-Lee. Mais uma vez, o Vogt foi superior com 45,41% e o Lower Bound o pior com 39,89%.

Juntando as quatro simulações é visível que adicionando o realismo do *capture effect* no ambiente as posições dos protocolos não se alteraram, mas diferentemente do caso sem *capture effect* onde a diferença do Vogt para o Eom-lee e do Eom-Lee para o Schoute era de aproximadamente 1%, aqui temos uma diferença do Vogt para o Eom-lee de aproximadamente 1,7% e do Eom-Lee para o Schoute de aproximadamente 0,4%.

1°	Vogt	45,7005641%
2°	Eom-Lee	44,0461694%
3°	Schoute	43,6112891%
4°	Lower Bound	40,7702231%

Figura 22 : Classificação da eficiência média dos protocolos *capture effect*.

Ao olhar esses resultados, uma coisa fica clara, o Eom-Lee que era absoluto como segundo melhor protocolo no ambiente teórico sem *capture effect* viu seu posto ameaçado pelo Schoute num ambiente realístico com o *capture effect*. Pode-se ver então claramente que o *capture effect* influencia cada protocolo de uma forma diferente.

3.2.3. Avaliação do Impacto do Capture Effect na Performance

Quando o *capture effect* é considerado vemos sempre uma melhora geral da *performance*. Isso acontece porque quando ele é considerado algumas colisões teóricas não acontecem na prática e acaba-se tendo uma maior taxa de sucessos [2].

O objetivo desta seção é ver como os protocolos se comportam diante do *capture effect* e qual a taxa de melhora que cada um deles tem.

Foram analisados todos os protocolos separadamente para os 4 tamanhos de *frames* iniciais já propostos. Todos os gráficos relacionados a esta seção podem ser encontrados no anexo C.

De todos os casos analisados, o que teve a melhor taxa de melhora foi o do protocolo Schoute para o *frame* inicial de 256 com uma melhora média de 37,38%.

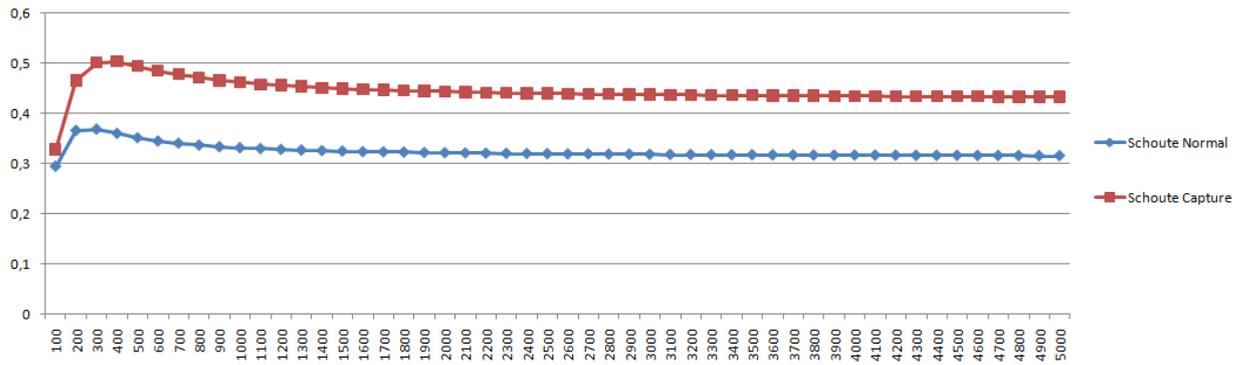


Figura 23: Grafico do impacto do capture effect na performande do Schoute.Caso do frame inicial sendo 256.

Enquanto o protocolo Eom-Lee para o mesmo *frame* inicial de 256 apresentou a pior melhora, uma média de apenas 31,94%.

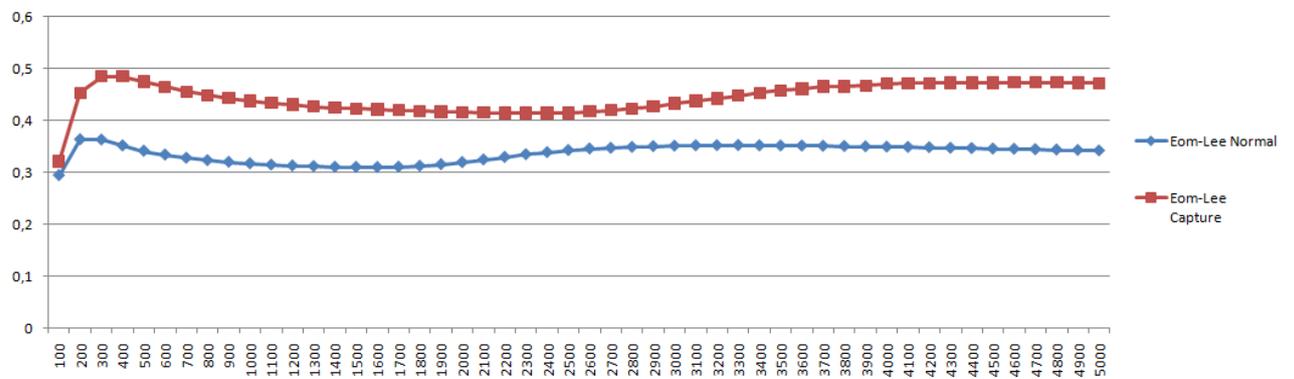


Figura 24: Grafico do impacto do capture effect na performande do Eom-Lee.Caso do frame inicial sendo 256.

Esses números já mostram a enorme diferença que pode existir entre a adaptação de um protocolo para o outro ao ambiente real. Para melhor decifrar esses efeitos foi feito uma média geral de melhora de *performance* para cada um dos protocolos. A seguir, temos 4 tabelas, cada uma medindo a eficiência média para um determinado protocolo.

Lower Bound			
Frame	Sem Captura	Com Captura	Melhora
32	0,294664673	0,398969528	35,39%
64	0,297172212	0,403454011	35,76%
128	0,301548111	0,410035433	35,97%
256	0,307143177	0,41834995	36,20%
Média Geral	0,300132043	0,407702231	35,83%

Figura 25: Tabela de eficiências e do impacto do capture do Lower Bound.

Schoute			
Frame	Sem Captura	Com Captura	Melhora
32	0,313510974	0,429224776	36,90%
64	0,315729188	0,432953693	37,12%
128	0,319118239	0,438104509	37,28%
256	0,323296485	0,444168587	37,38%
Média Geral	0,317913722	0,436112891	37,17%

Figura 26: Tabela de eficiências e do impacto do capture do Schoute.

Vogt			
Frame	Sem Captura	Com Captura	Melhora
32	0,334654587	0,454153628	35,70%
64	0,333494556	0,455193303	36,49%
128	0,337804607	0,458602543	35,75%
256	0,337093759	0,460073091	36,48%
Média Geral	0,335761877	0,457005641	36,11%

Figura 27: Tabela de eficiências e do impacto do capture do Vogt.

Eom-Lee			
Frame	Sem Captura	Com Captura	Melhora
32	0,32751126	0,428323942	30,78%
64	0,321072442	0,438942532	36,71%
128	0,32100073	0,451508393	35,73%
256	0,335802694	0,44307191	31,94%
Média Geral	0,326346782	0,440461694	33,79%

Figura 28: Tabela de eficiências e do impacto do capture do Eom-Lee.

Com as médias de melhora de eficiência com o *capture effect* dos quatro protocolos pode-se fazer uma comparação para ver como o protocolo Eom-Lee está diante dos outros três protocolos.

1°	Schoute	37,17%
2°	Vogt	36,11%
3°	Lower Bound	35,83%
4°	Eom-Lee	33,79%

Figura 28: Classificação do impacto do *capture* nos protocolos.

Vendo a comparação entre os quatro protocolos vemos claramente que o Eom-Lee não responde bem ao efeito *capture effect*, o que prejudica seu rendimento na prática e mostra claramente que ele não foi criado pensando-se no efeito que o *capture effect* teria sobre a *performance* do sistema como um todo.

3.3.Considerações

3.3.1. Performance de Reconhecimento

Diferente do sugerido em [15], o protocolo Eom-Lee não possui melhor *performance* em relação à quantidade de *slots* necessárias para reconhecer todas as etiquetas de um sistema. O resultado das simulações em 3.2.1 e 3.2.2 mostram claramente que o protocolo Vogt ainda é o protocolo que necessita no geral de menos *slots* para reconhecer as etiquetas de um sistema.

Não se pode deixar de ressaltar, no entanto, que o protocolo Eom-Lee é um protocolo valioso e novo e que em alguns casos chega até a ser superior ao Vogt e com um custo computacional relativamente menor.

3.3.2. Capture Effect

Como foi mostrado na seção 2.4.4, ao criar o protocolo e deduzir as fórmulas estatísticas, [15] considera um ambiente ideal onde se duas etiquetas transmitem ao mesmo tempo, uma colisão com certeza irá acontecer. Ao tomar essa premissa como verdadeira ele desconsiderou o ambiente real e criou um protocolo que funciona muito bem num ambiente ideal mas que, como mostrado na seção 3.2.3, tem a pior adaptação de todos ao ambiente real.

Esse trabalho também mostra que [3] estava correto ao considerar e analisar os efeitos do efeito de captura afirmando que podiam alterar significativamente o efeito de um

protocolo. Também mostra que a consideração contrária de [2], de que o *capture effect* não teria um grande impacto na *performance* de um protocolo estava errada.

4. Conclusão

Esse trabalho mostrou um estudo sobre o funcionamento de protocolos anti-colisão em sistemas RFID. O foco do trabalho foi mostrar as vantagens e desvantagens do novo protocolo Eom-Lee em comparação a protocolos já consolidados como o Lower Bound, o Schoute e o Vogt.

As principais contribuições deste trabalho foram a demonstração de que o protocolo Vogt continua sendo o protocolo de melhor *performance* da atualidade e foi mostrado também que o efeito de captura pode mexer significativamente na *performance* de um protocolo, não podendo portanto ser ignorado no momento de construção do mesmo.

Para chegar a esta conclusão foram vistos todos os conceitos básicos sobre sistemas RFID, sobre protocolos anti-colisão, sobre os quatro protocolos estudados neste trabalho e sobre o efeito de captura.

Para avaliar a *performance* do Eom-Lee tentamos nos aproximar ao máximo do ambiente real com a introdução de posicionamento de etiquetas em relação ao leitor, potência de sinal e o efeito de captura.

Através de diversas simulações diferentes, viu-se que o protocolo de melhor *performance* da atualidade continua sendo o Vogt, seja no campo teórico ou na prática.

Neste trabalho foi provado também que o efeito de captura, algo ignorado por grande parte dos autores, pode influenciar significativamente o desempenho de um protocolo e foi mostrado como o Eom-Lee é um dos protocolos que têm uma das piores adaptações ao *capture effect* prejudicando assim sua *performance* em ambientes reais.

Como um trabalho futuro, pretendemos trabalhar na melhora da *performance* do protocolo Eom-Lee. O Eom-Lee tem um custo computacional relativamente menor que o Vogt e portanto necessita de menos trabalho do *hardware*. Por isso enxergamos nesse protocolo, um protocolo que apesar de ainda não ser o de melhor *performance*, um protocolo que pode ser melhorado para se tentar chegar a tal situação.

5. Referências

- [1] D. Braun. "RFID: Etiqueta inteligente conquista território do código de barras" Computer World. 03/10/2006. <http://computerworld.uol.com.br/telecom/2006/10/03/idgnoticia.2006-10-03.6009072159/> Acessado: 10/09/2010
- [2] H. Vogt. "Efficient Object Identification with Passive RFID Tags" in Proc. International Conference on Pervasive Computing, pp. 98-113, 2002.
- [3] B. Li, Y. Yang e J. Wang. "Anti-Collision Issues analysis in Gen2 Protocol". Auto-ID Labs White Paper WP-HARDWARE-047. 03/2009
- [4] A. C. Martins. "Design e Implementação de Sistema Pervisível e Vestível". Trabalho de graduação do Cin-UFPE. 18/06/2010
- [5] S. Gaudin. "RFID market to hit \$9.7 billion by 2013". Computer World. 21/05/2008 http://www.computerworld.com/s/article/9087760/Researcher_RFID_market_to_hit_9.7_billion_by_2013?taxonomyId=11&taxonomyName=development Acessado: 25/08/2010
- [6] R.A. Silva. "Um Sistema de Auxílio à Localização de Etiquetas RFID em ambientes Internos". Dissertação de Mestrado do Cin-UFPE. 22/09/2008
- [7] A. F. CUNHA. "RFID – Etiquetas com eletrônica de ponta-RFID" Revista Saber Eletrônica 401. http://www.techtraining.eng.br/conteudo/Artigo-RFID-Parte_II.PDF. Acessado: 15/11/2010
- [8] K. Finkenzerler. *RFID Handbook, Fundamentals and Applications in Contactless Smart Cards and Identification*. John Wiley and Sons Ltd., 2003.
- [9] B.A. Jesus, "Um protocolo híbrido de anti-colisão de etiquetas para sistemas RFID". Dissertação de Mestrado do Cin-UFPE. 03/2010.
- [10] Law, K. Lee, and K.-Y. Siu, "Efficient memoryless protocol for tag identification (extended abstract)," in Proceedings of the 4th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, (Toronto, CA), pp. 75–84, Aug. 2000.
- [11] S. Piramuthu. Adaptive Framework for Collisions in RFID Tag Identification. Journal of Information & Knowledge Management, 7(1), pp. 9–14, May, 2008.
- [12] Wen-Tzu Chen and Guan-Hung Lin. An Efficient Anti-Collision Method for Tag Identification in a RFID System. IEICE Transactions on Communications, Vol. E89-B, No.12, pp. 3386–3392, December 2006.
- [13] F.C. Schoute, "Dynamic frame length ALOHA," IEEE Trans. Commun. Vol.31, No. 4, pp. 565-568, Apr. 1983.

[14] D. K. Klair, K. W. Chin, and R. Raad, "*On the accuracy of RFID tag estimation functions*," in Proc. IEEE ISCT, pp. 1401-1406, Oct. 2007.

[15] J. Eom and T.-J. Lee, "Accurate Tag Estimation For Dynamic Framed Slotted ALOHA in RFID Systems" IEEE Communication Letters Vol. 14, No. 1, January 2010.

[16] Theodore S. Rappaport, "Wireless Communications – Principles & Practice," *Prentice Hall Communications Engineering and Emerging Technologies Series*, 1996.

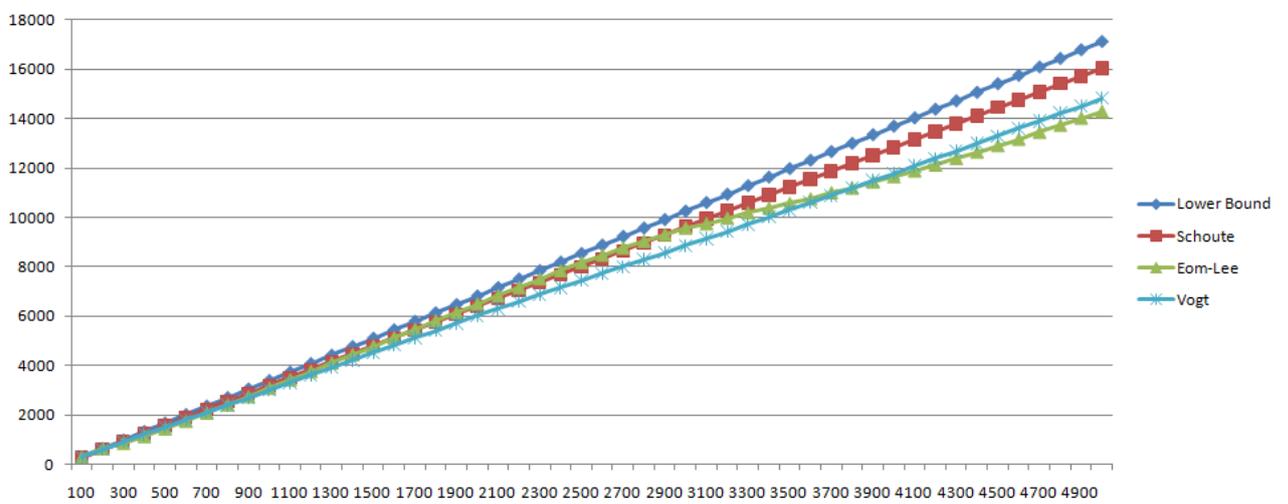
[17] S. Lee, S. Joo and C. Lee, "An Enhanced Dynamic Framed Slotted ALOHA Algorithm for RFID Tag Identification," in Proc. Of MobiQuitous, pp.166 - 172, Jul. 2005.

Anexos

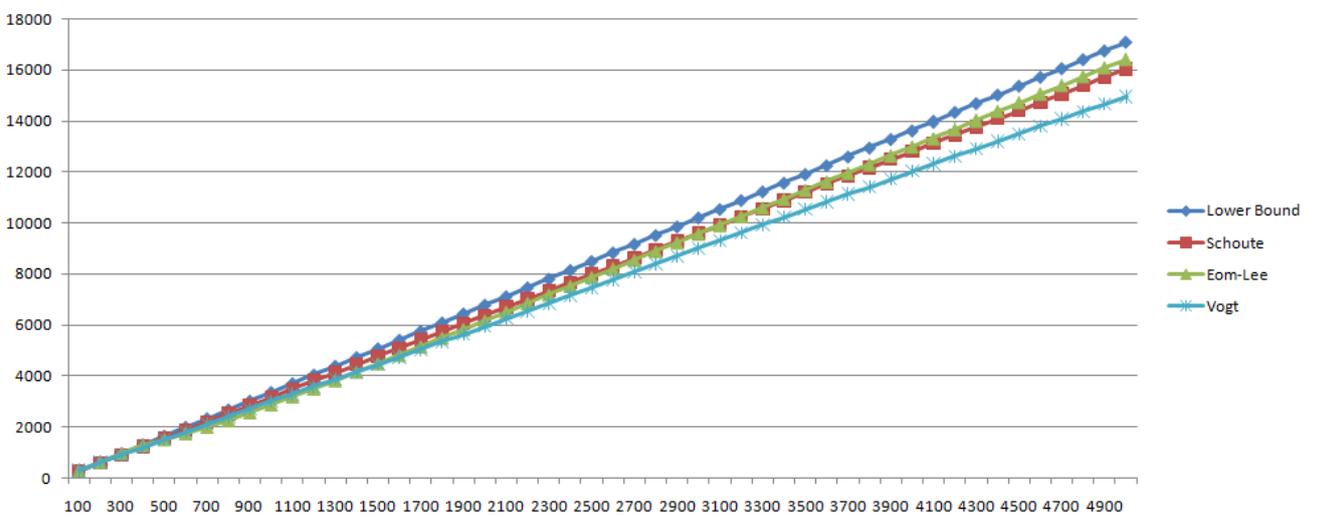
Anexo A – Resultado das Simulações – Slots x Tags sem o efeito de captura

Esse tipo de simulação avaliou quantos *slots* cada protocolo utilizou para ler todas as etiquetas de um sistema. Foram feitas simulações sem o efeito de captura.

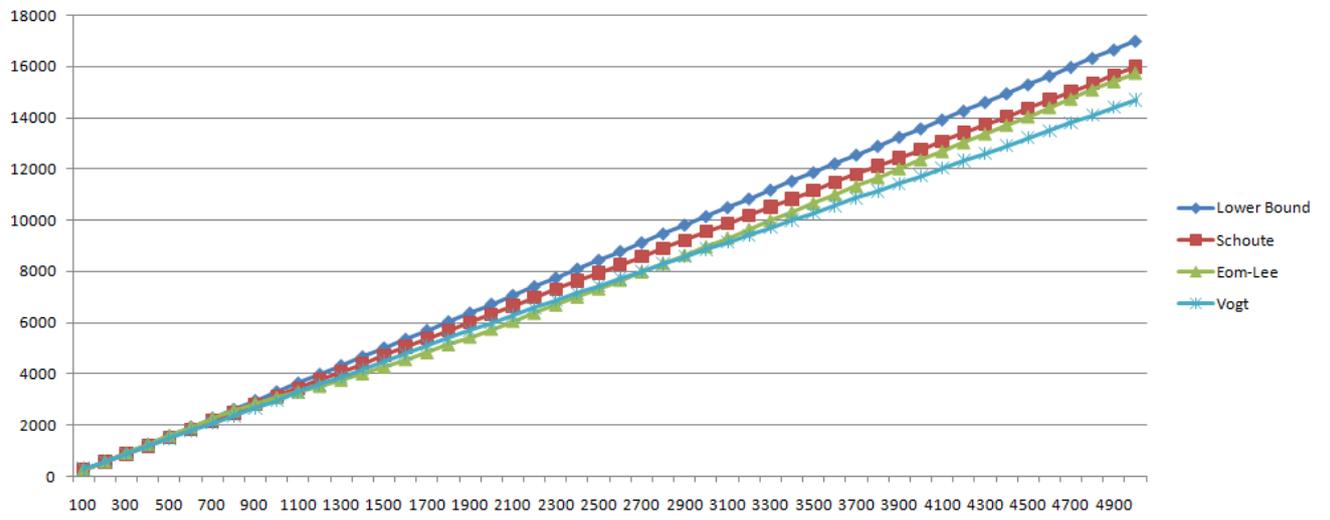
Frame Inicial 32



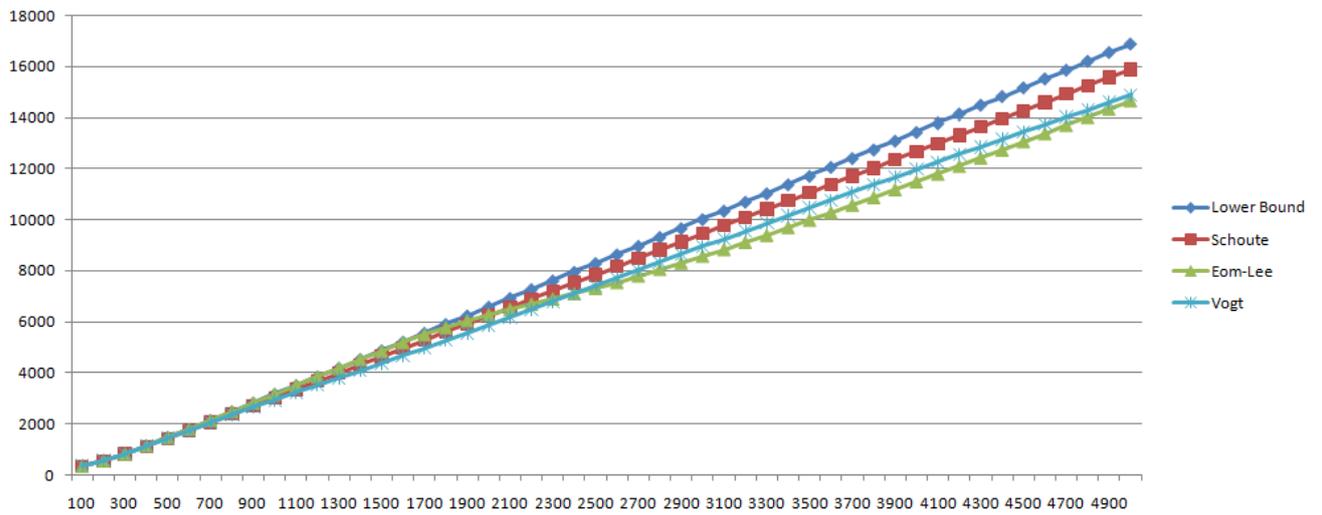
Frame Inicial 64



Frame Inicial 128



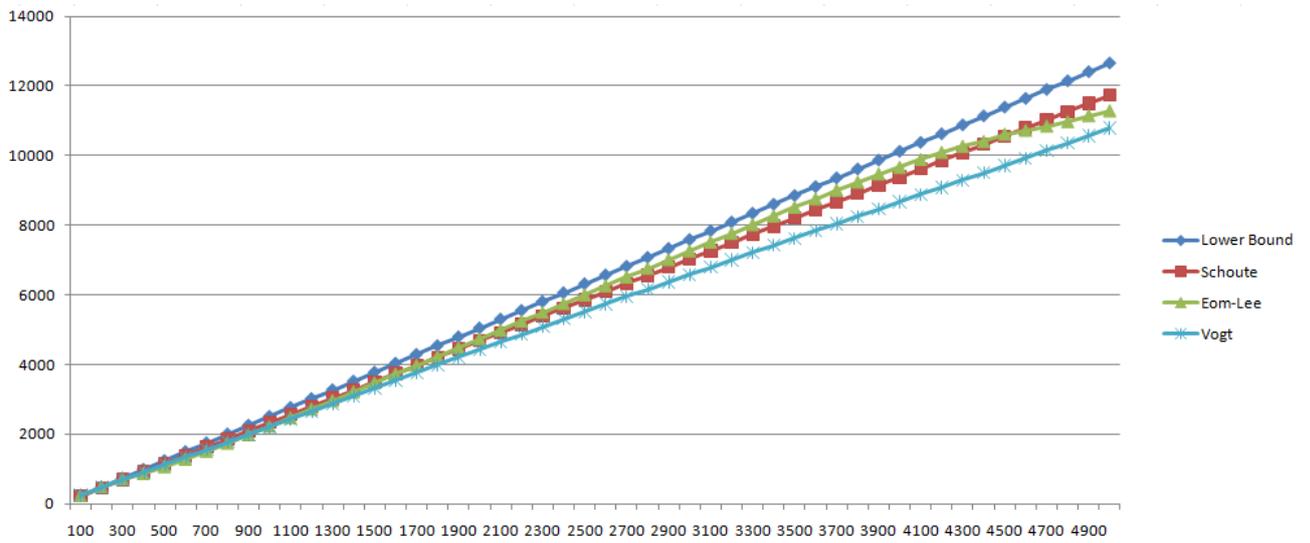
Frame Inicial 256



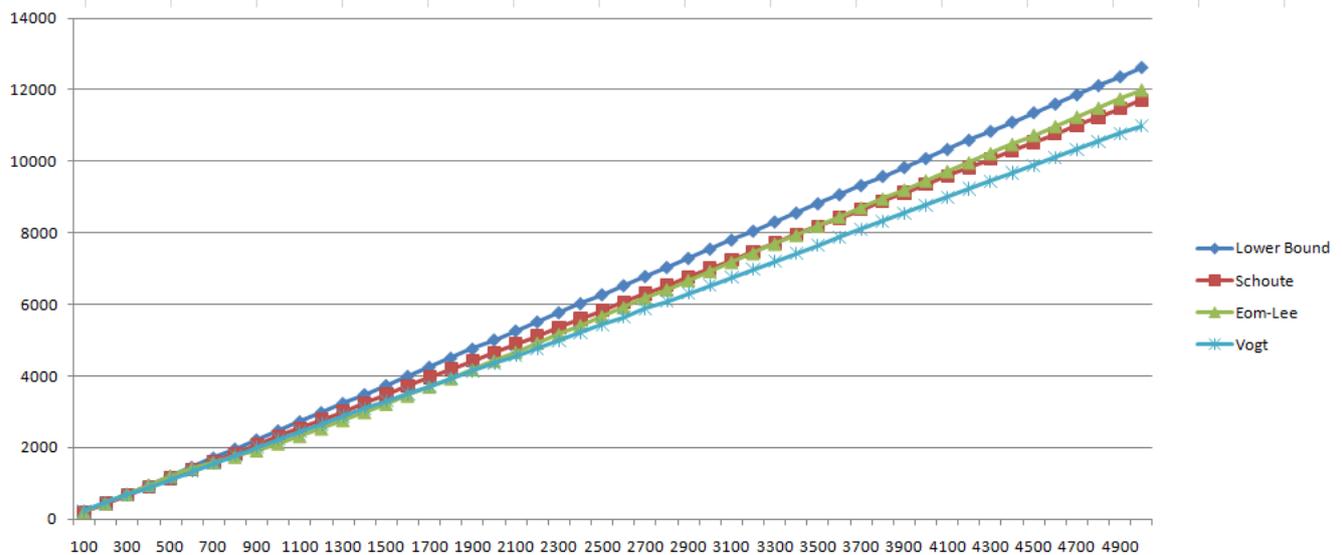
Anexo B – Resultado das Simulações – Slots x Tags com o efeito de captura

Esse tipo de simulação avaliou quantos *slots* cada protocolo utilizou para ler todas as etiquetas de um sistema. Foram feitas simulações com o efeito de captura.

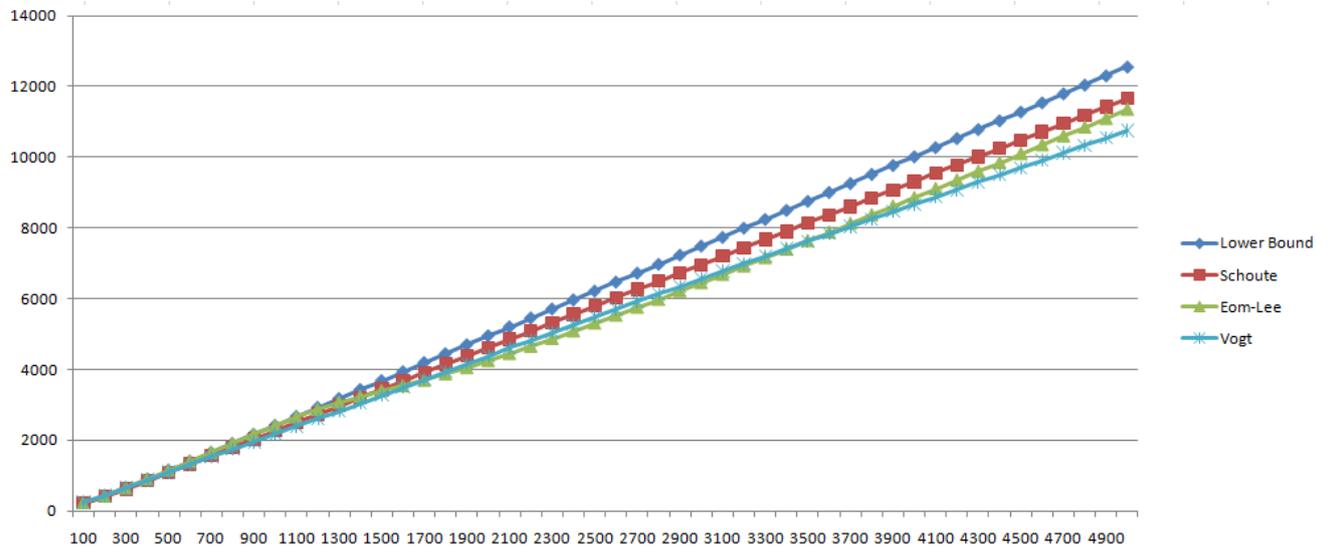
Frame Inicial 32



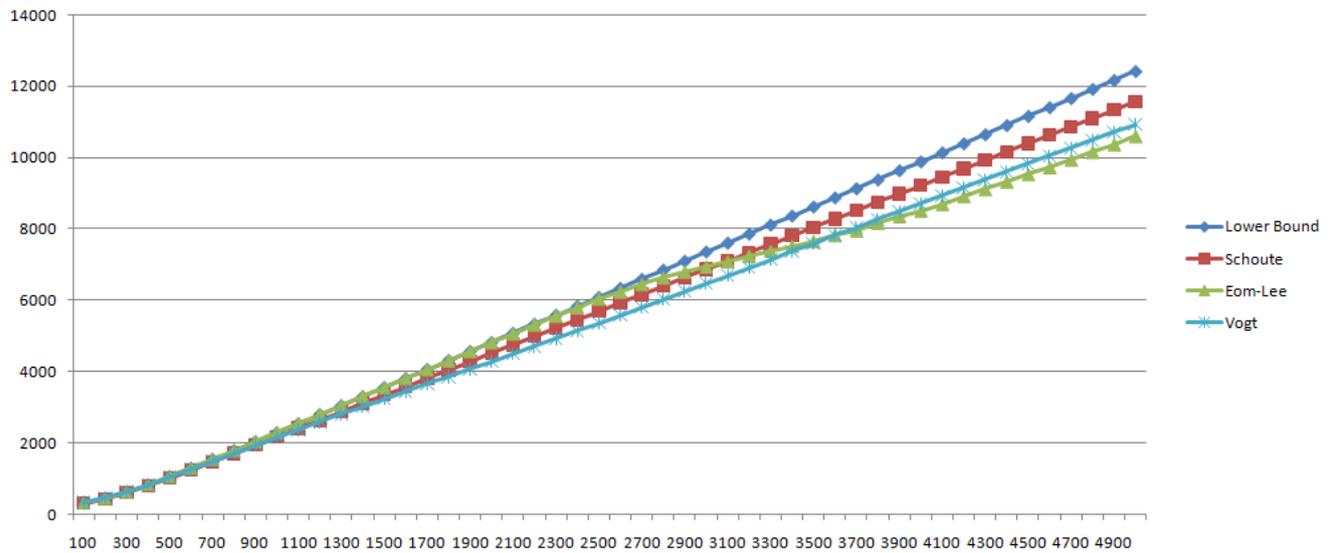
Frame Inicial 64



Frame Inicial 128

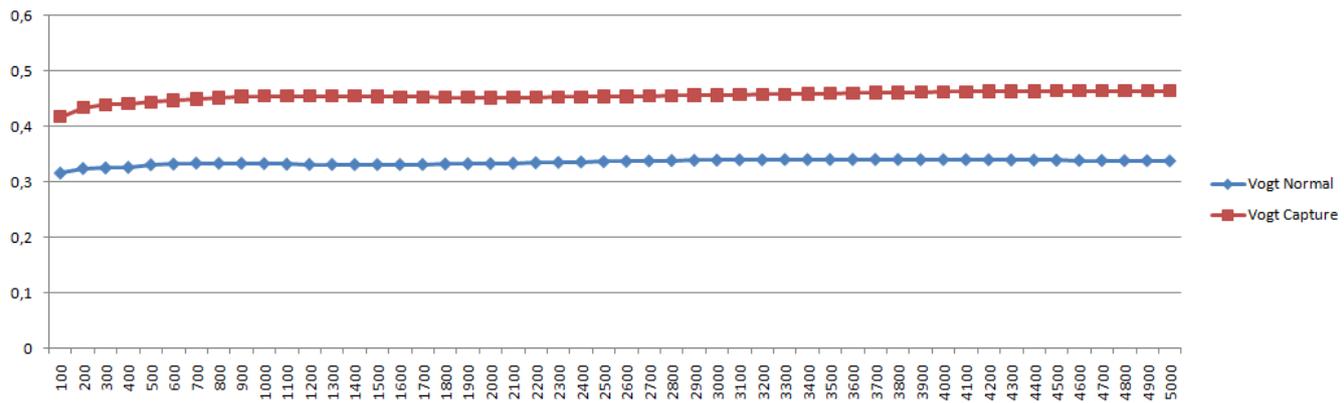
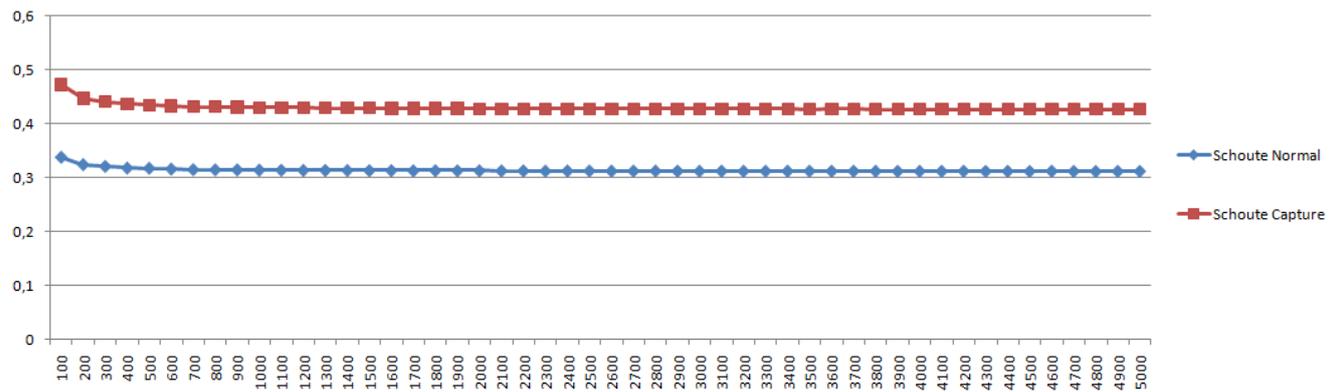
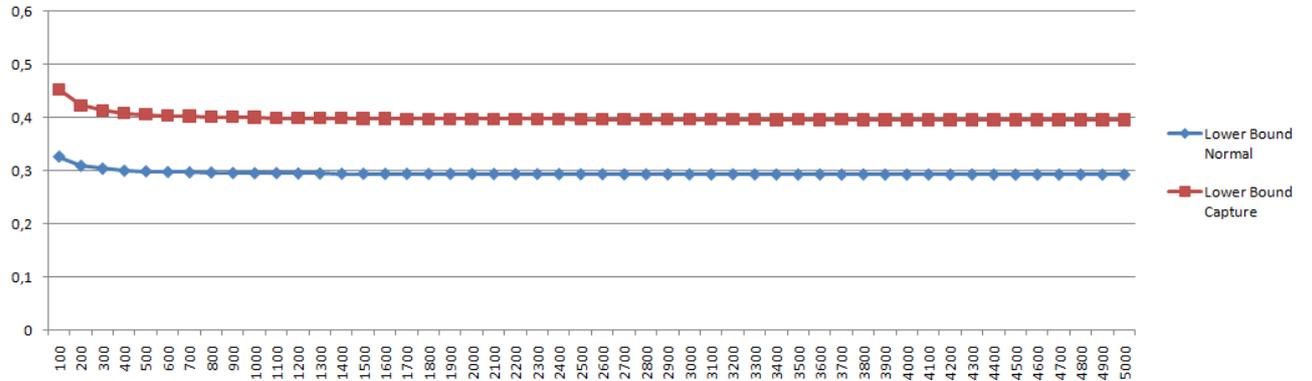


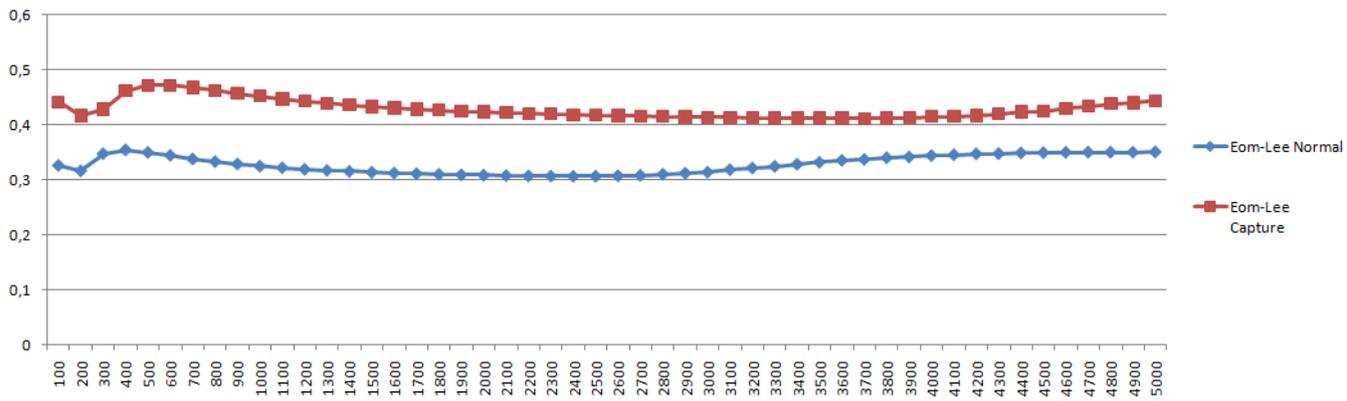
Frame Inicial 256



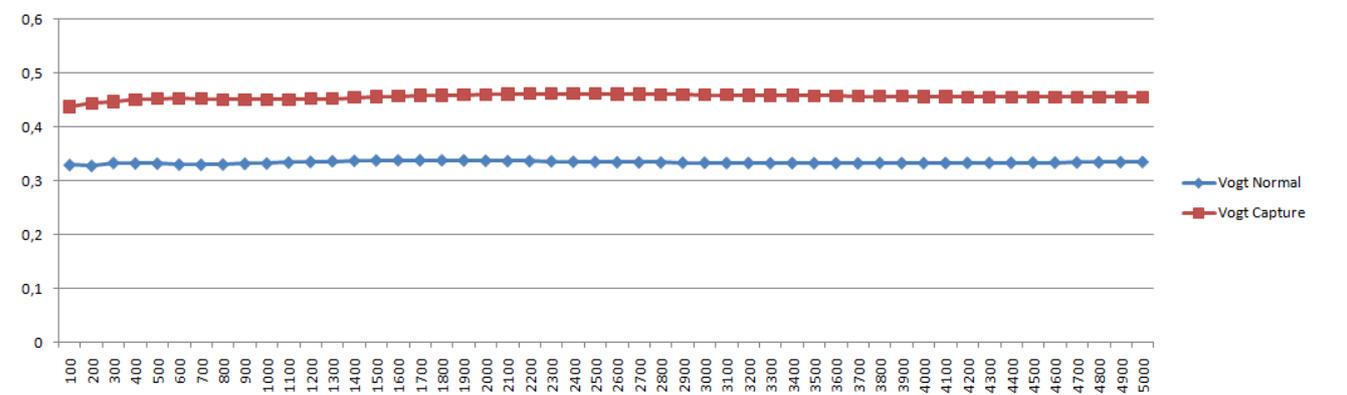
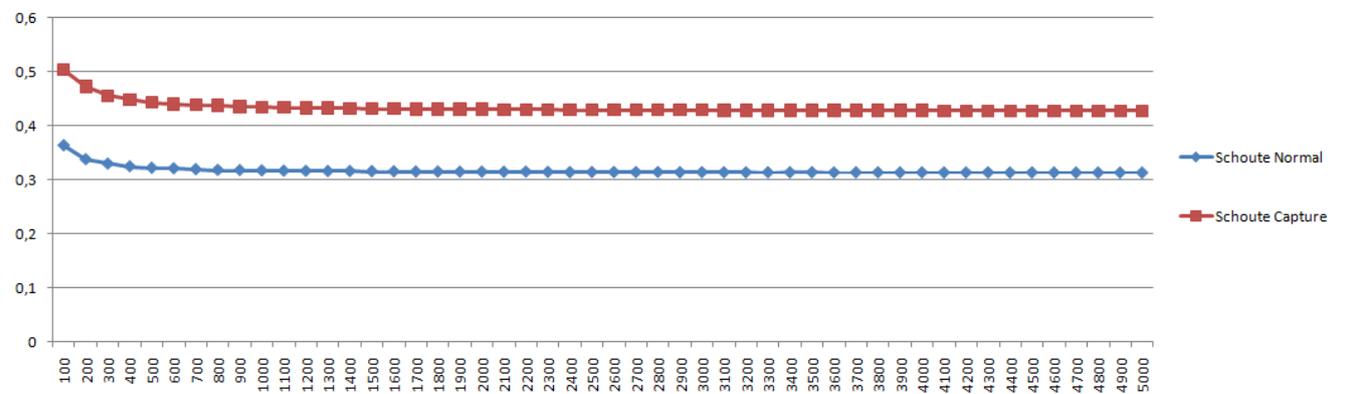
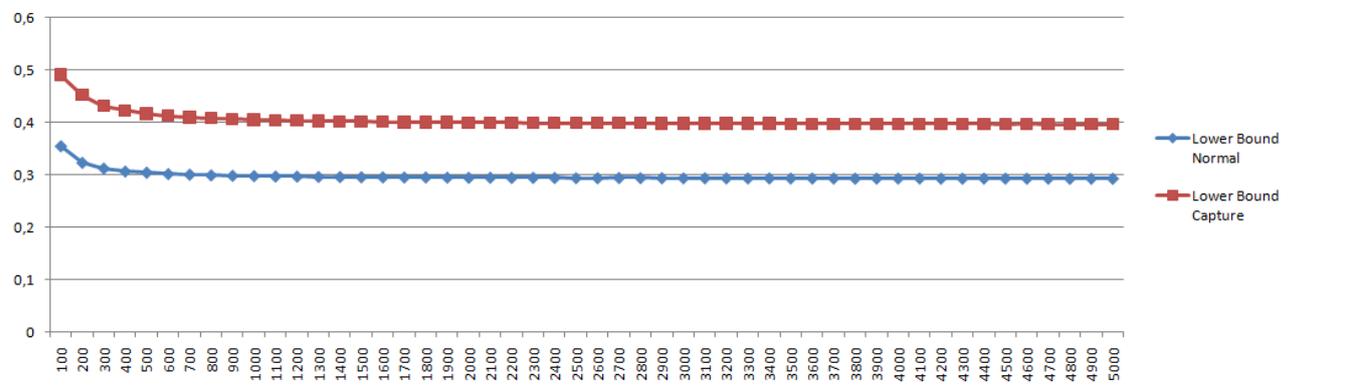
Anexo C – Resultado das Simulações – Eficiência Com Capture Effect Vs Eficiência Sem Capture Effect

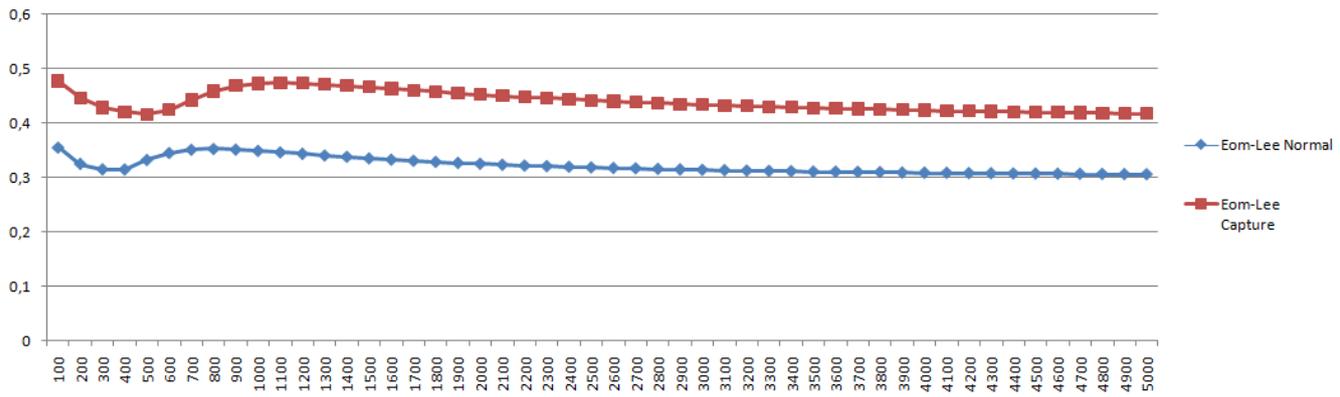
Frame Inicial 32



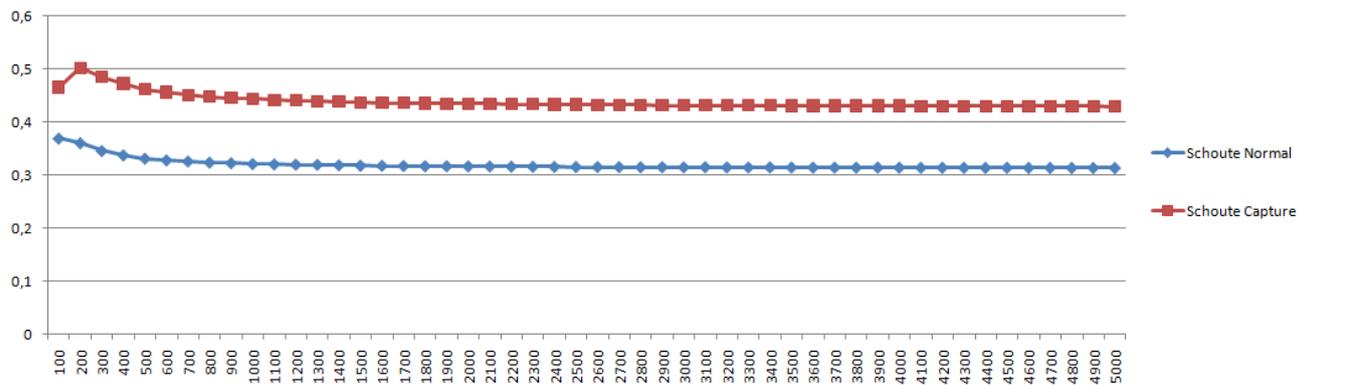
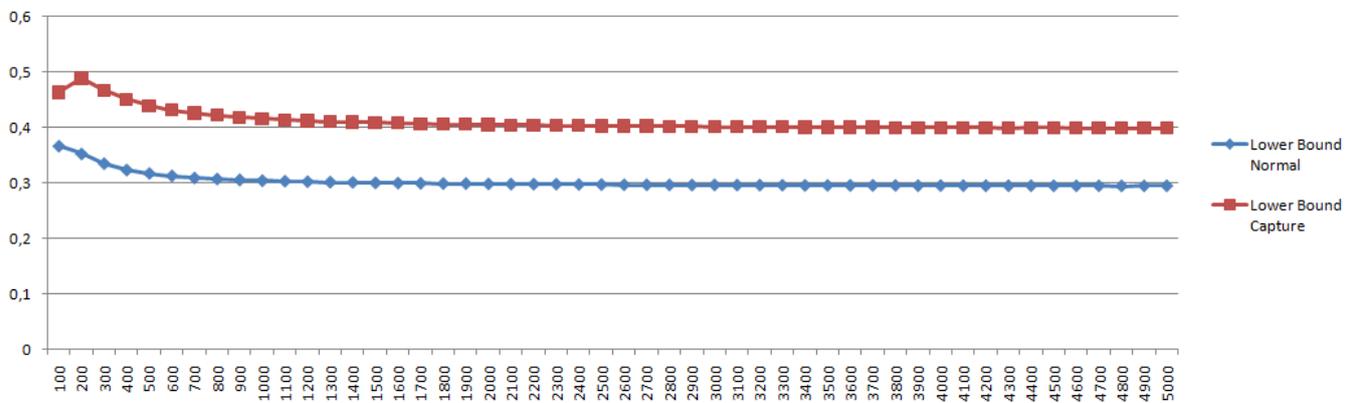


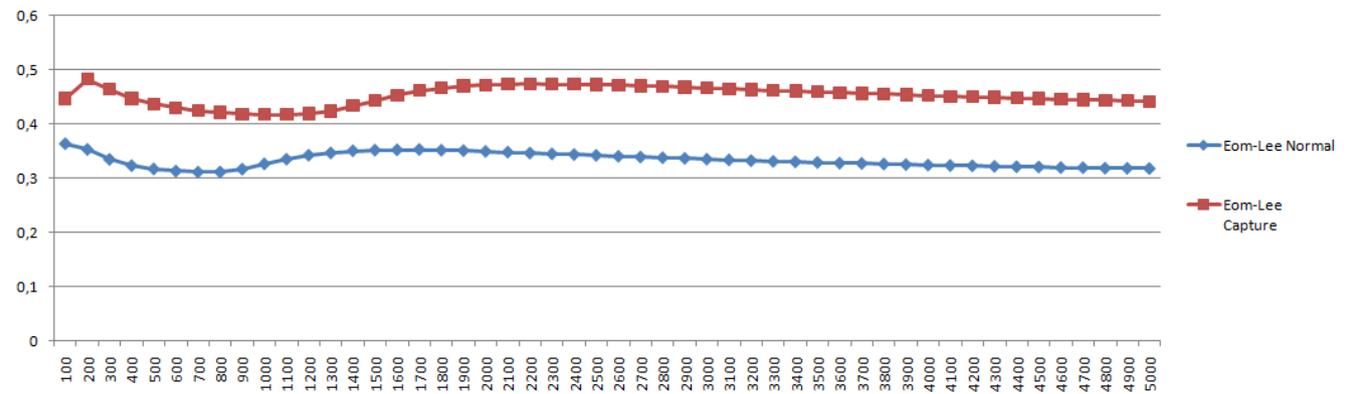
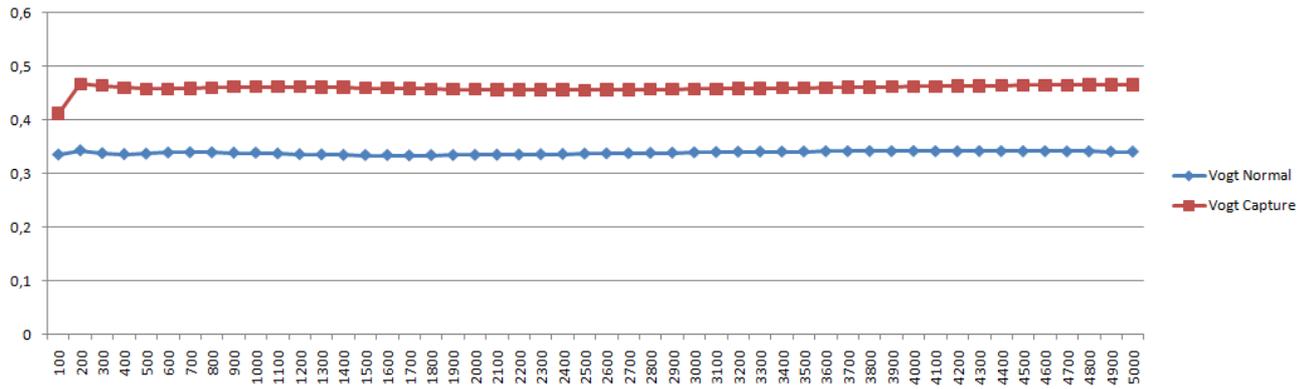
Frame Inicial 64





Frame Inicial 128





Frame Inicial 256

