

O Problema do Acesso Múltiplo ao Meio em Sistemas RFID

Módulo II-B



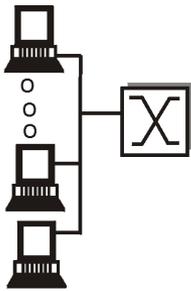
PROTOSCOLOS DE ACESSO MÚLTIPLO AO MEIO DE COMUNICAÇÃO (REVISÃO)

Enlaces e Protocolos de Acesso Múltiplo



Dois tipos de “links”:

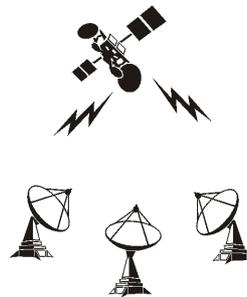
- Ponto-a-ponto (point-to-point)
 - PPP para acesso dial-up
 - Enlace ponto-a-ponto entre switch Ethernet e host
- **broadcast** (meio compartilhado)
 - Ethernet tradicional
 - upstream HFC (Hybrid Fiber-Coaxial)
 - LAN sem fio IEEE 802.11
 - RFID



shared wire
(e.g. Ethernet)



shared wireless
(e.g. Wavelan)



satellite



cocktail party

Protocolos de Acesso Múltiplo



- Canal broadcast compartilhado único
- 2 ou mais transmissões simultâneas: interferência
 - Há colisão se nó recebe 2 ou mais sinais ao mesmo tempo

Protocolo de acesso múltiplo

- Algoritmo distribuído que determina como os nós compartilham o canal, i.e., determina quando o nó pode transmitir
- Comunicação sobre compartilhamento de canal deve usar o próprio canal!
 - Não há canal *fora de banda* (out-of-band) para coordenação de transmissões

Protocolo de Acesso Múltiplo Ideal



Canal broadcast com taxa de R bps

1. Quando 1 nó deseja transmitir, ele envia dados à taxa R .
2. Quando M nós desejam transmitir, cada um pode transmitir a uma taxa média R/M
3. Completamente descentralizado:
 - Não há nó especial para coordenar transmissões
 - Não há sincronização de relógios, slots
4. Simples

Protocolos MAC: taxonomia



3 classes:

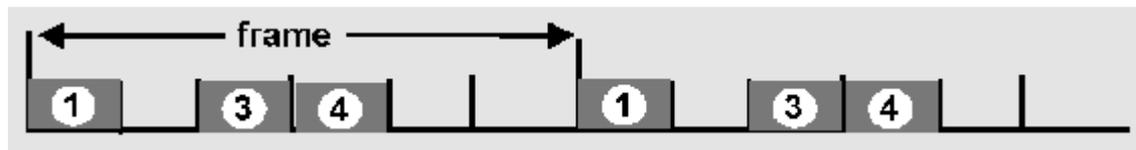
- Particionamento de Canal
 - divide canal em pequenas “partes” (slots de tempo, frequência, código)
 - aloca “parte” para uso exclusivo do nó
- Acesso Randômico
 - canal não é dividido e podem ocorrer colisões
 - Precisa tratar colisões
- “Taking turns”
 - Cada nó aguarda sua vez para transmitir. Um passa a vez para o outro. Nós que desejam enviar mais experimentarão maiores atrasos

Particionamento de Canal: TDMA



TDMA: time division multiple access

- Acesso ao canal ocorre em "rounds"
 - Cada host recebe um slot de tamanho fixo em cada round
 - slots não utilizados são desperdiçados (ficam idles ou vazios)
 - exemplo: 6 hosts em um LAN, 1,3,4 possuem pkt, slots 2,5,6 idle (vazios)
-
- TDM (Time Division Multiplexing): canal dividido em N slots de tempo, 1 por usuário; ineficiente com usuários com baixo ciclo de trabalho e a altas cargas.

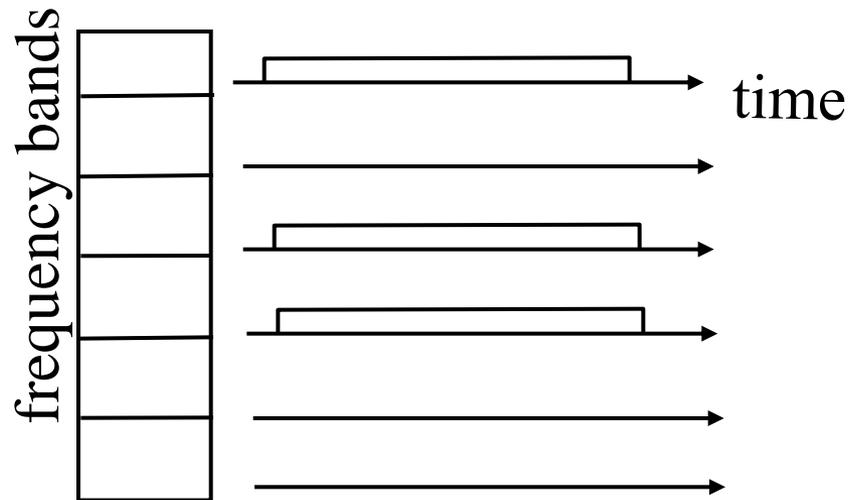


Particionamento de Canal: FDMA



FDMA: frequency division multiple access

- Espectro do canal dividido em bandas de frequência
 - Cada host recebe uma banda de frequência
 - Se não há transmissões na banda, ela é desperdiçada
 - Exemplo: 6 hosts em um LAN, 1,3,4 possuem pkt, bandas 2,5,6 sem uso
-
- FDM (Frequency Division Multiplexing): a frequência é subdividida.



Protocolos de Acesso Randômico



- Quando nó possui pacote para enviar
 - Transmite a taxa máxima R do canal.
 - Não há coordenação prévia entre nós
- 2 ou mais nós transmitindo ao mesmo tempo → “colisão”,
- **Protocolos MAC de acesso randômico** especificam:
 - Como detectar colisões
 - Como tratar colisões (e.g., atrasando retransmissões)
- Exemplos de protocolos MAC de acesso randômico:
 - slotted ALOHA
 - ALOHA
 - CSMA, CSMA/CD, CSMA/CA

Slotted ALOHA



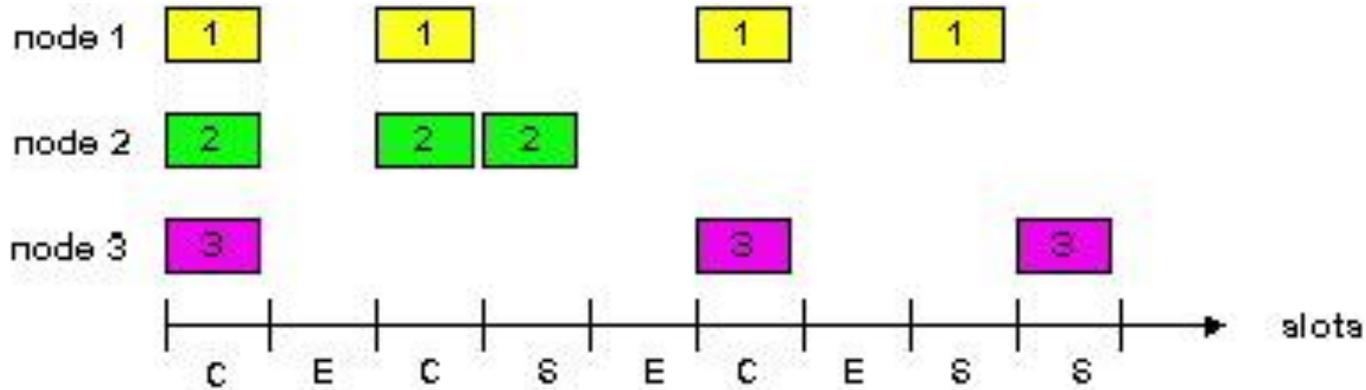
Hipóteses

- frames de mesmo tamanho
- Tempo é dividido em slots de mesmo tamanho, tempo para transmitir um quadro da camada enlace
- Nós iniciam transmissão somente no início de um slot
- Nós estão sincronizados
- se 2 ou mais nós transmitem em um mesmo slot, todos detecam a colisão

Funcionamento

- Quando nó obtém um novo quadro da camada enlace para transmitir, ele o transmite no próximo slot
- Se não há colisão, nó pode enviar novo quadro da camada enlace no slot seguinte
- Se há colisão, nó retransmite quadro da camada enlace em cada slot subsequente com probabilidade p até o sucesso

Slotted ALOHA



Prós

- Um único nó ativo pode transmitir continuamente na taxa máxima do canal
- Altamente descentralizado: somente slots precisam estar sincronizados
- simples

Contras

- colisões, desperdício de slots
- Slots livres
- Nós precisam ter a habilidade de detectar colisões em um tempo menor que o necessário para transmitir um pacote
- sincronização

Eficiência do Slotted Aloha



Eficiência é a fração de slots utilizados com sucesso a longo-prazo quando há muitos nós, cada qual com muitos quadros a serem transmitidos

- Suponha N nós com muitos quadros a serem enviados, cada nó transmite em um slot com probabilidade p
- Probabilidade de que 1 nó tenha sucesso em um slot = $p(1-p)^{N-1}$
- Probabilidade de que qualquer nó tenha sucesso = $Np(1-p)^{N-1}$

- Para obter a eficiência máxima com N nós, encontre p^* que maximiza $Np(1-p)^{N-1}$
- Quando há muitos nós, o limite de $Np^*(1-p^*)^{N-1}$ quando N vai a infinito, indica uma eficiência máxima de $1/e = .37$

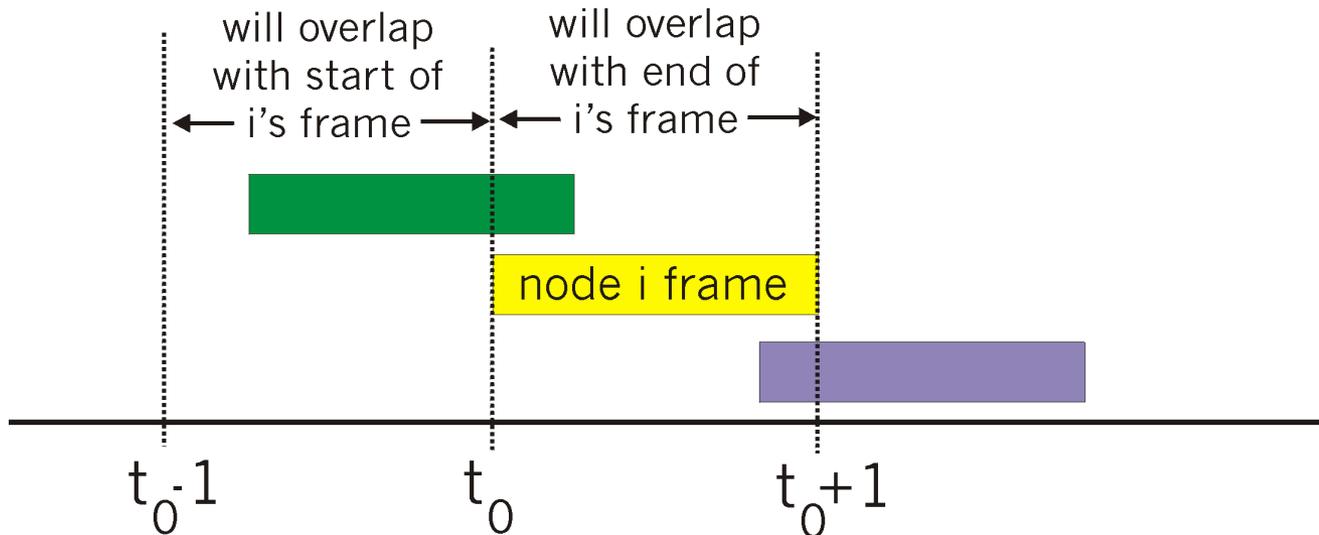
No melhor dos casos:

Canal usado em transmissões úteis 37% do tempo!

ALOHA Puro (unslotted)



- unslotted Aloha: mais simples, sem sincronização
- Quando camada enlace possui quadros para transmitir
 - transmite-os imediatamente
- Probabilidade de colisão aumenta:
 - Quadro enviado em t_0 colide com outros quadros enviados em $[t_0-1, t_0+1]$



Aloha Puro: Eficiência



$P(\text{sucesso de um dado nó}) = P(\text{nó transmitir}) \cdot$

$P(\text{nenhum outro nó transmitir em } [t_0-1, t_0]) \cdot$

$P(\text{nenhum outro nó transmitir em } [t_0, t_0+1])$

$$= p \cdot (1-p)^{N-1} \cdot (1-p)^{N-1}$$

$$= p \cdot (1-p)^{2(N-1)}$$

... escolhendo p ótimo e fazendo $N \rightarrow$ infinito ...

Ainda pior! $= 1/(2e) = .18$

CSMA (Carrier Sense Multiple Access)

CSMA: escuta antes de transmitir:

Se canal está “idle” (vazio): transmite o quadro completo

- Se canal está ocupado, adiar transmissão

- Analogia humana: não interrompa os outros!

Colisões no CSMA



colisões ainda podem ocorrer:

Ter atraso de propagação significa que dois nós podem não ouvir a transmissão um do outro

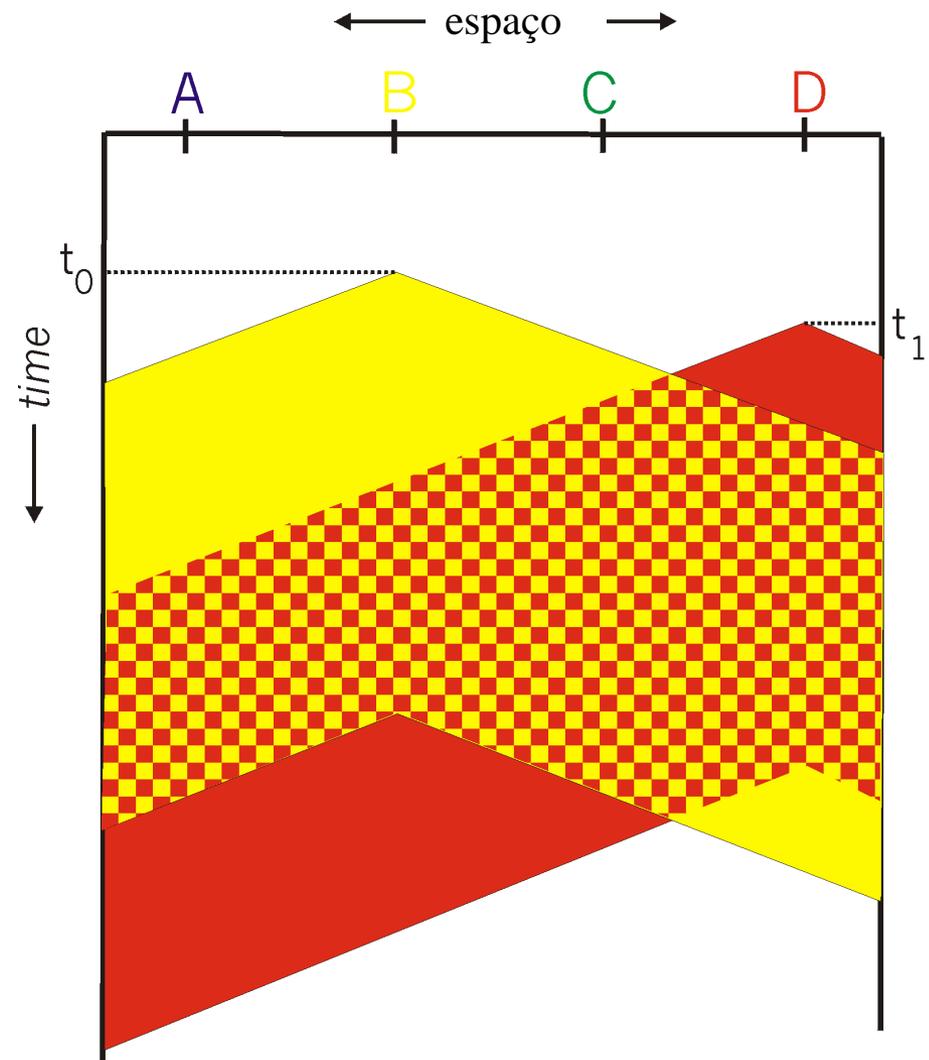
colisão:

Tempo total de transmissão do pacote é desperdiçado

nota:

distância & atraso de propagação são importantes para se determinar a probabilidade de colisão

Distribuição espacial dos nós



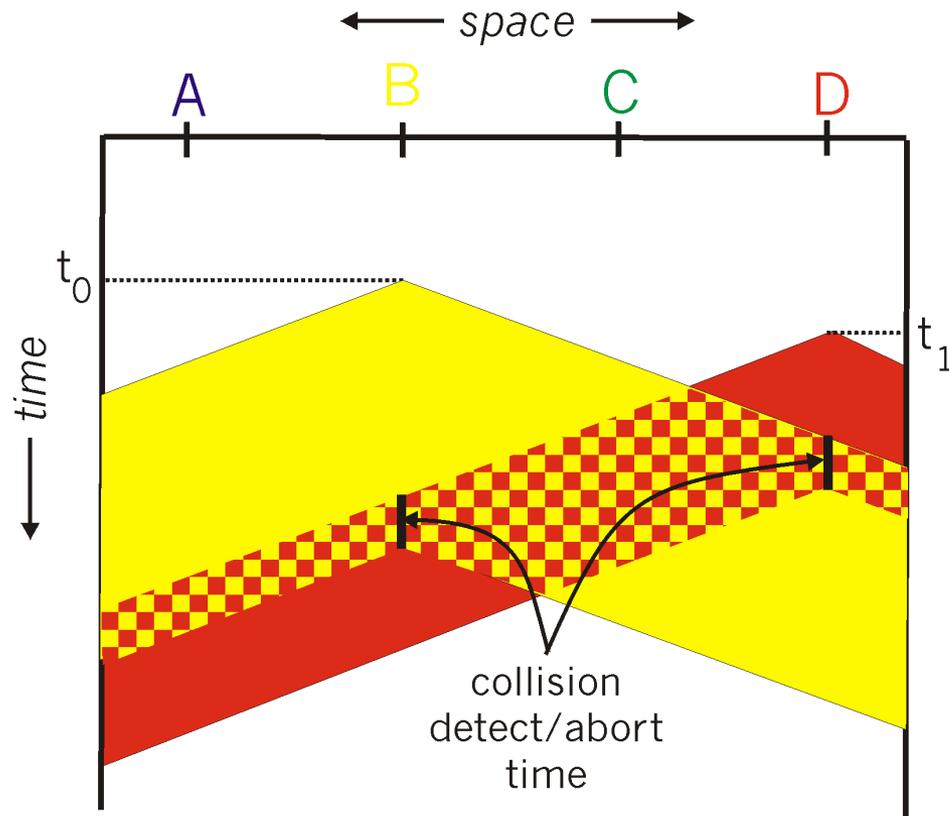
CSMA/CD (Collision Detection)



CSMA/CD: escuta portadora, adiamento como no CSMA

- colisões *detectadas* em um intervalo curto de tempo
 - Transmissões colidindo são abortadas, reduzindo o desperdício do canal
- Detecção de colisão:
 - Fácil em LANs cabeadas: mede força de sinais, compara força do sinal transmitido com o recebido
 - Difícil em LANs sem fio: receptor desliga enquanto se transmite. Só receptor sabe se houve colisão ou não
 - Analogia humana: uma pessoa educada com grande habilidade de conversação

CSMA/CD (collision detection)



Protocolo MAC “Taking Turns”



Protocolos MAC de Particionamento de Canal

- Compartilhamento eficiente do canal e justo com alta carga
- Ineficiente com baixa carga: atraso de acesso ao canal, $1/N$ da banda alocada mesmo se somente 1 nó estiver ativo!

Protocolos MAC de Acesso Randômico

- Eficiente com baixa carga: um único nó pode usar toda a capacidade do canal
- Alta carga: sobrecarga (overhead) de colisões

Protocolos “taking turns”

Olham para o melhor de ambos os mundos!

Protocolo MAC “Taking Turns”

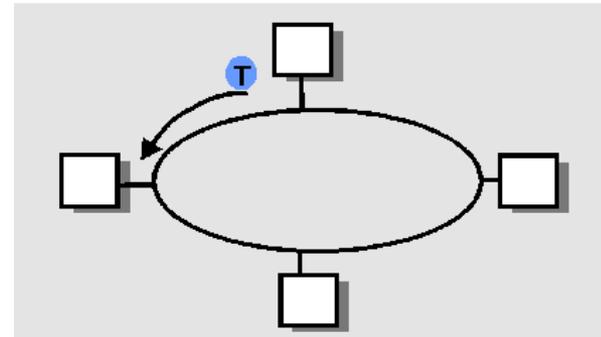


Polling:

- Nó mestre “convida” nós escravos a transmitirem e convite segue uma ordem ou seqüência
- fraquezas:
 - polling overhead
 - latência
 - Ponto único de falha (mestre)

Passagem de Token:

- ❑ **Token** de controle passado de um nó a outro sequencialmente
- ❑ Mensagem token
- ❑ fraquezas:
 - token overhead
 - latência
 - ponto único de falha (token)



Resumo dos protocolos MAC



- O que fazer com um meio compartilhado?
 - Particionamento de canal, no tempo, na frequência ou por código
 - Time Division, Frequency Division
 - Particionamento Randômico (dinâmico),
 - ALOHA, S-ALOHA, CSMA, CSMA/CD
 - Escuta de portadora: fácil com algumas tecnologias (cabeadas), difícil em outras (sem fio)
 - CSMA/CD usado no Ethernet
 - CSMA/CA usado no IEEE 802.11 (Wi-Fi)
 - Taking Turns
 - polling feito por um coordenador, passagem de token



ACESSO MÚLTIPLO AO MEIO COM SISTEMAS RFID

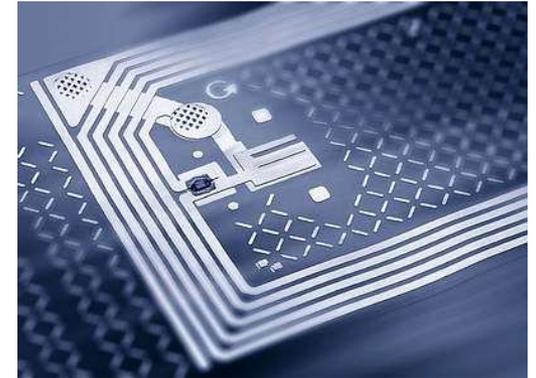
Motivação



- **Sistemas RFID** (Radio Frequency Identification)

- **Arquitetura**

- **Etiquetas**
 - Passivas (recebem energia do leitor / alcance de comunicação de até vários metros)
 - Armazenam ID único
- **Leitores**
- **Base de Dados**
- **Servidor**

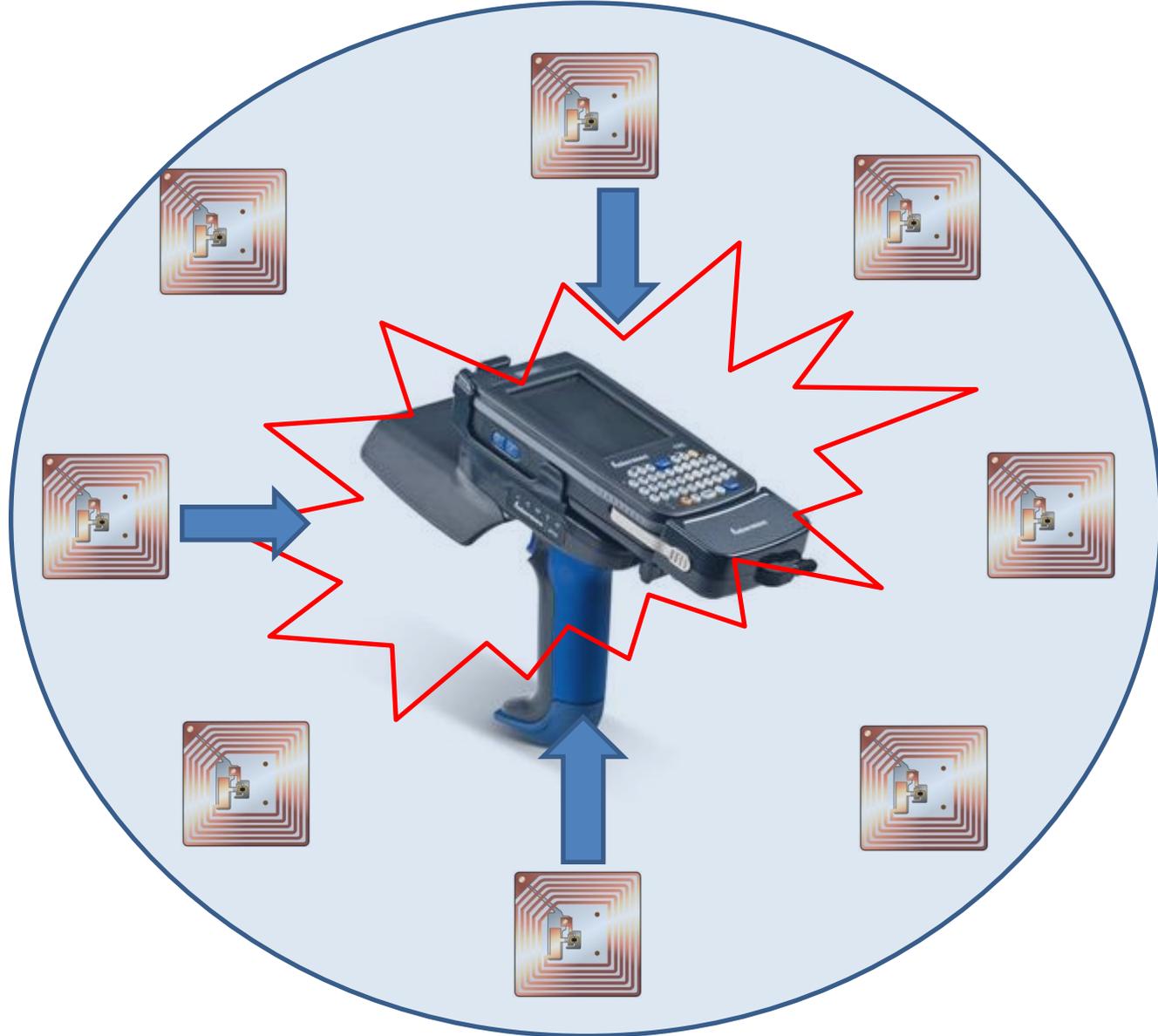


Etiqueta



Leitor

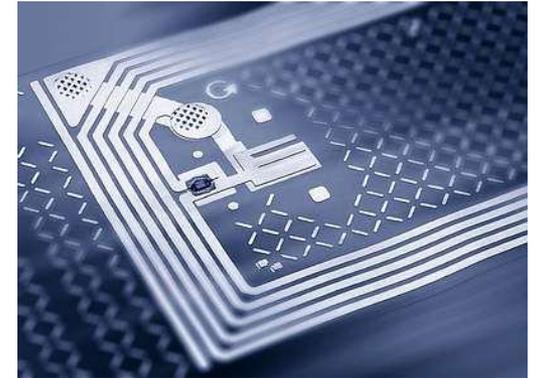
Motivação



Motivação



- Problema de **acesso múltiplo ao meio** de comunicação
- Requer **solução diferenciada** para sistemas RFID
- Etiquetas Passivas
 - **Poucos recursos** computacionais e de memória
 - **Incapazes** de detectar colisões e de escutar transmissões de outras etiquetas



Etiqueta

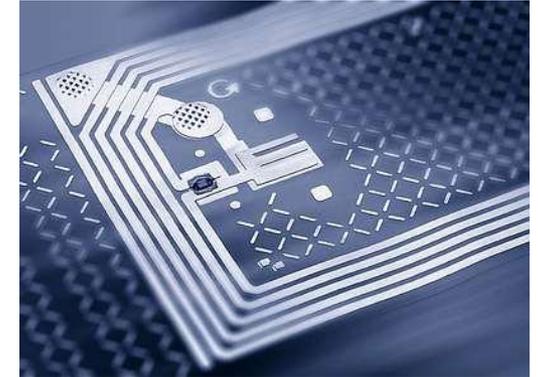


Leitor

Motivação



- Solução?
 - **Acesso ao meio** de comunicação deve ser arbitrado pelo leitor
- Como?
 - Através de um **protocolo anticolisão**
 - Existem vários: baseados em árvore, baseados em ALOHA e híbridos ... mas vamos nos concentrar agora no DFSA (**D**ynamic **F**ramed **S**lotted **A**LOHA)



Etiqueta

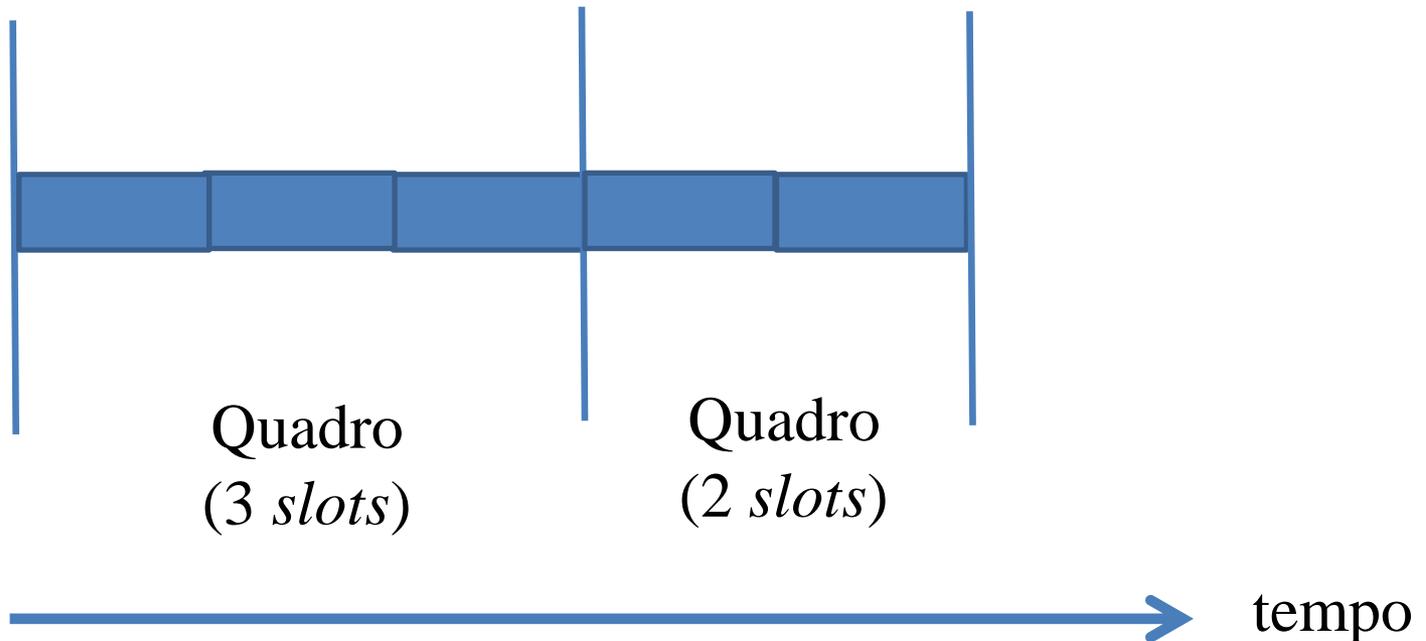


Leitor

Motivação



- **DFSA** (Dynamic Framed Slotted ALOHA)
 - **Leitor organiza tempo** em um ou mais quadros que são subdivididos em *slots* de tempo

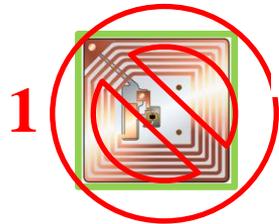
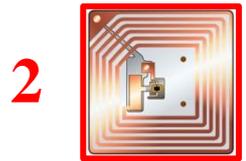
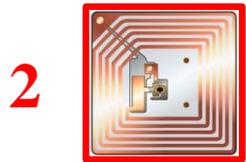
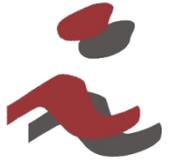


Motivação

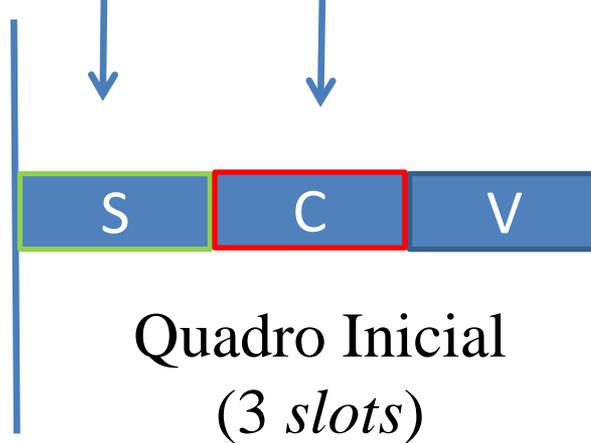


- **DFSA** (Dynamic Framed Slotted ALOHA)
 - **Leitor requisita etiquetas a transmitirem** em um *slot* por quadro até serem identificadas
 - Tamanho de cada **quadro** subsequente ao inicial é **ajustado dinamicamente**

Motivação



silenciamento



S: Sucesso
C: Colisão
V: Vazio

Motivação

- Ainda restam etiquetas a serem identificadas ...
- Mas qual o **tamanho ideal** para o próximo quadro?



Motivação

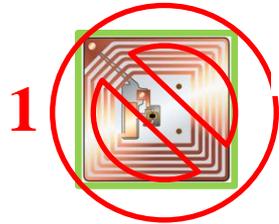
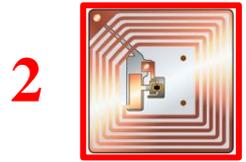
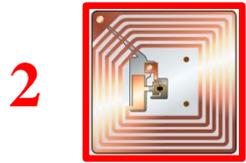
- Quando os *slots* possuem o mesmo tamanho ...

O **tamanho ideal** do próximo quadro para **maximizar a eficiência** deve ser **igual ao número de etiquetas restantes (backlog)**



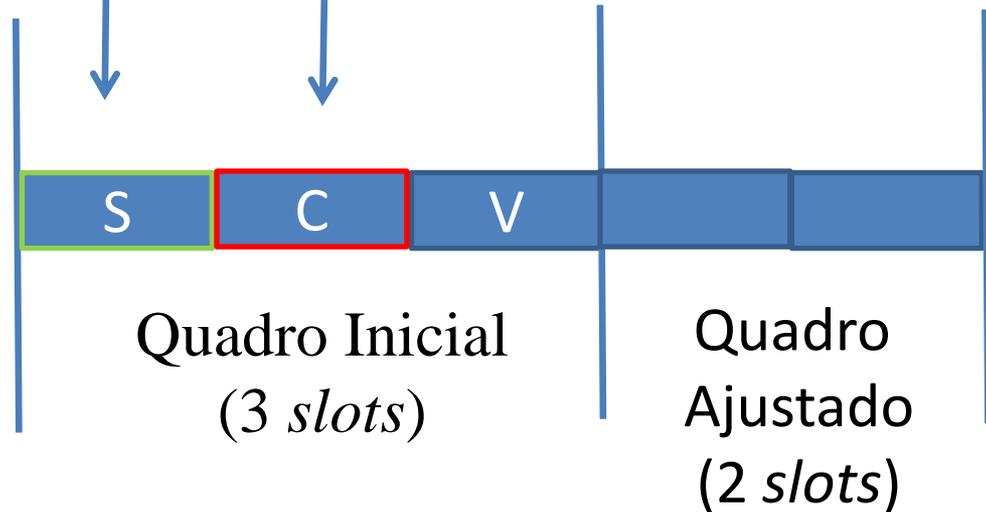
Fácil não?

Motivação

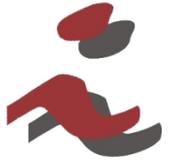


silenciamento

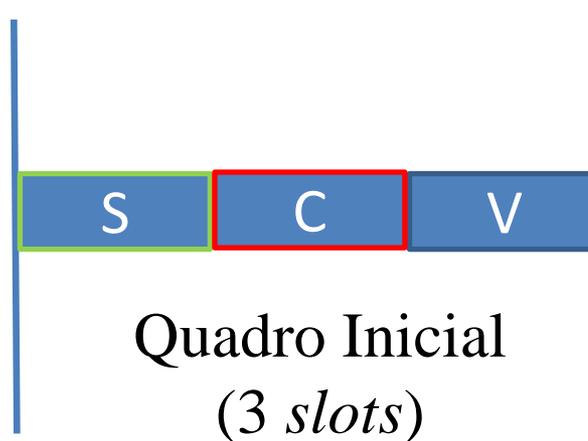
S: Sucesso
C: Colisão
V: Vazio



Motivação

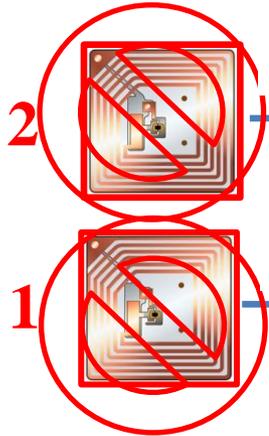


S: Sucesso
C: Colisão
V: Vazio



Quadro
Ajustado
(2 slots)

Motivação

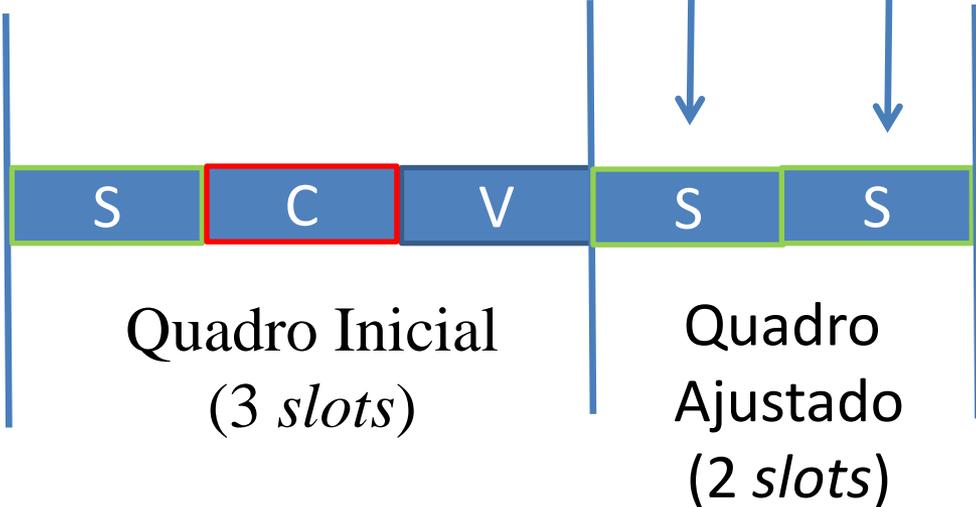


silenciamento

silenciamento

S: Sucesso
C: Colisão
V: Vazio

Fim do Processo



Motivação

- Quando os *slots* possuem o mesmo tamanho ...

O **tamanho ideal** do próximo quadro para **maximizar a eficiência** deve ser **igual ao número de etiquetas restantes (backlog)**

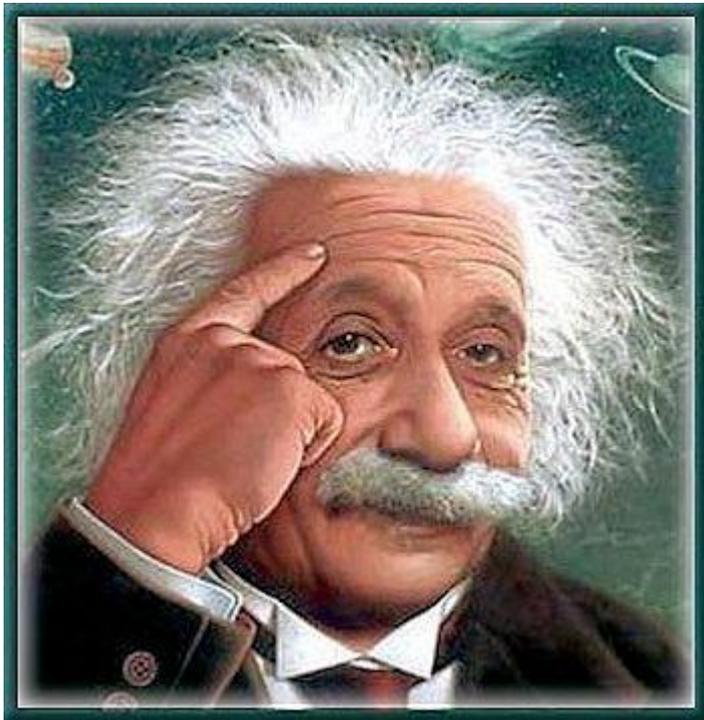


Mas como o leitor conhece o backlog?



Motivação

- Solução ...



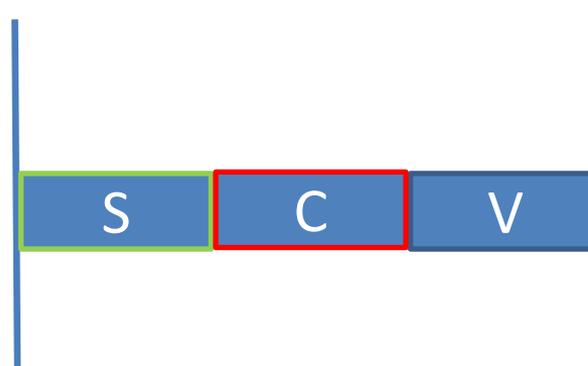
Usar um estimador !



Motivação

- Após o término de um quadro com ao menos 01 colisão, **o leitor executa um estimador**
- Estimação da população de etiquetas que competiram por *slots* no quadro analisado é feita **com base na quantidade de *slots* vazios**, bem sucedidos ou em colisão

S: Sucesso
C: Colisão
V: Vazio

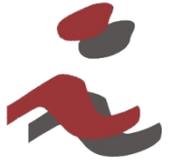




ESTIMADORES PARA O DFSA

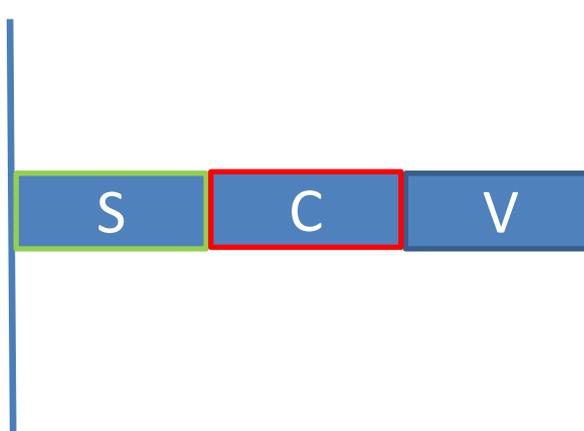


Estimador Lower Bound



- Qual **a menor quantidade possível** de etiquetas envolvidas em uma colisão?

S: Sucesso
C: Colisão
V: Vazio



Duas!

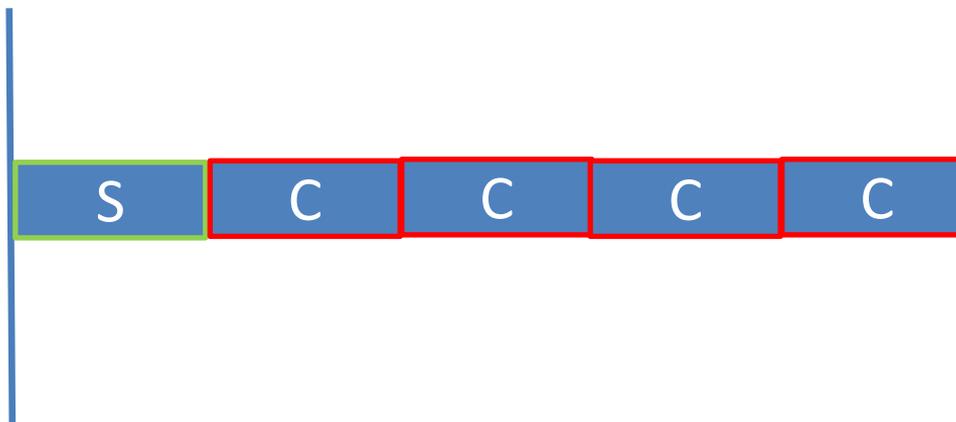


Estimador Lower Bound



- Quantas etiquetas **teriam competido** por *slots* neste exemplo?

S: Sucesso
C: Colisão
V: Vazio



$$1 + (2 \times 4) = 9$$

Estimador Lower Bound



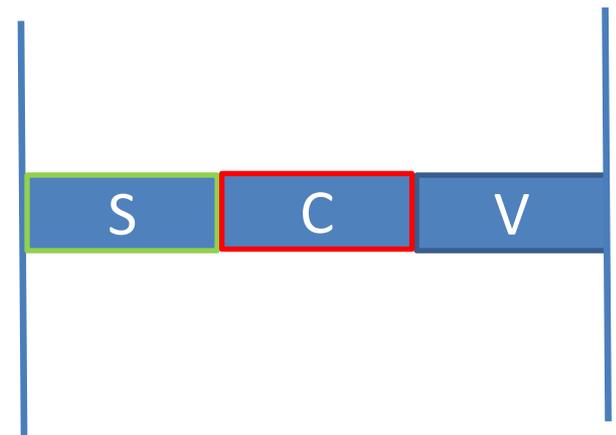
- Então ... o que o Lower Bound define?

$$\hat{n} = s_s + 2 \cdot s_c$$

Estimativa de etiquetas que competiram por *slots*

$$\hat{f} = 2 \cdot s_c$$

Tamanho do Próximo Quadro



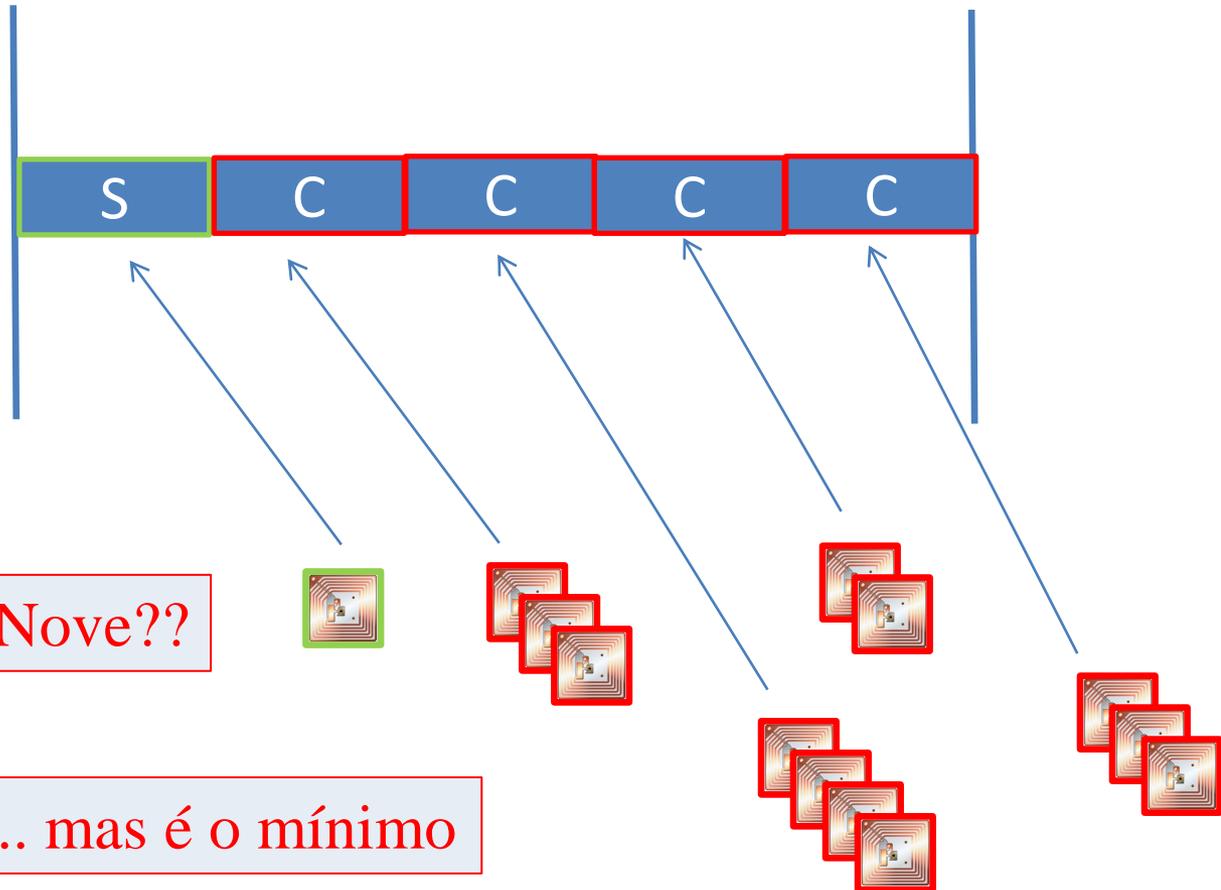
Estmador Lower Bound



- É um bom estimador?

$$\hat{n} = s_s + 2 \cdot s_c$$

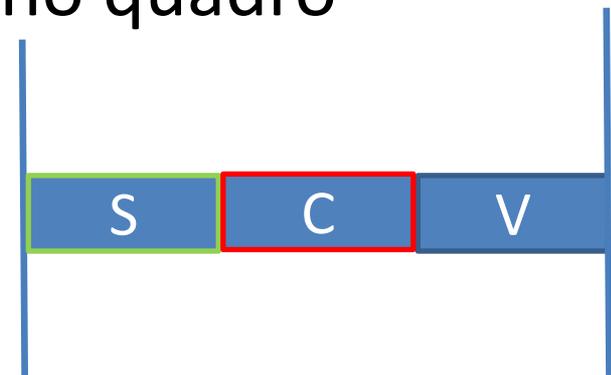
S: Sucesso
C: Colisão
V: Vazio



Estimador Schoute



- Considera um **processo de chegadas do tipo Poisson**
 - Ajuda a estudar sequência de eventos aleatórios que ocorrem ao longo do tempo
- Encontra o **número esperado de etiquetas** que transmitirão no próximo quadro
- Esse número é igual a **2,39** multiplicado pelo **número total de slots em colisão** no quadro analisado



Estimador Schoute



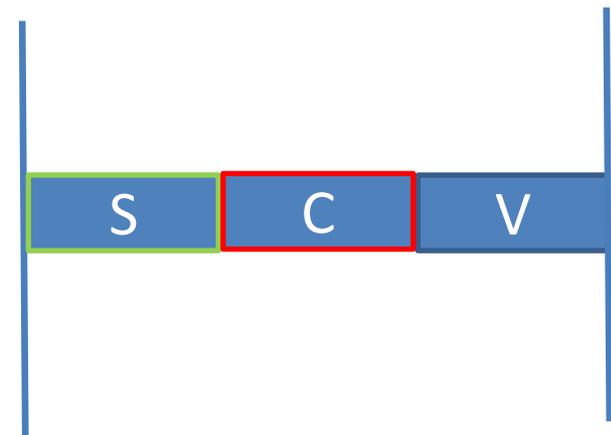
- Então ... o que o Schoute define?

$$\hat{n} = s_s + 2,39 \cdot s_c$$

Estimativa de etiquetas que competiram por *slots*

$$\hat{f} = 2,39 \cdot s_c$$

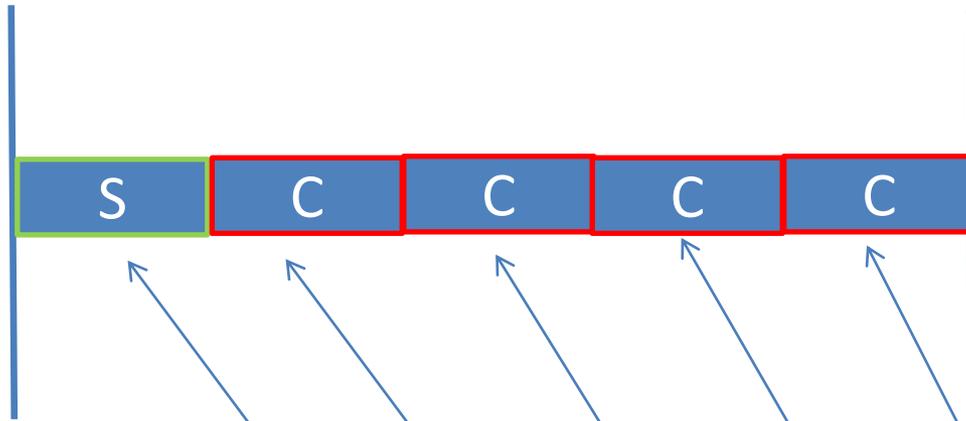
Tamanho do Próximo Quadro





Errei para menos!

S: Sucesso
C: Colisão
V: Vazio



$$1 + (4 \times 2,39) = 10,56$$

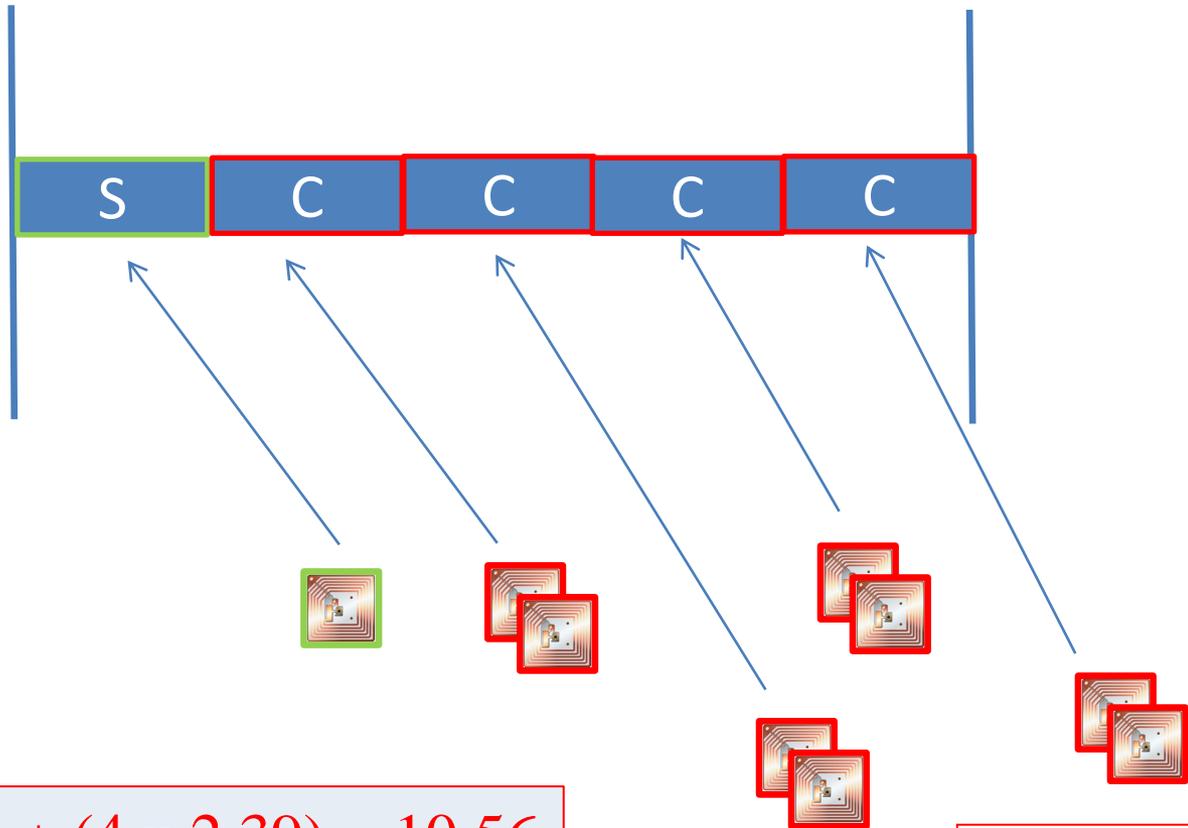
$$\hat{n} = s_s + 2,39 \cdot s_c$$

Estimador Schoute



Errei para mais!

S: Sucesso
C: Colisão
V: Vazio



$$1 + (4 \times 2,39) = 10,56$$

$$\hat{n} = s_s + 2,39 \cdot s_c$$

Problemática dos Estimadores



Errei para menos!

- Aumenta potencialmente o número de *slots* em colisão
- Requer mais quadros
- Requer mais tempo

Errei para mais!

- Aumenta potencialmente o número de *slots* vazios
- Os quadros ficam maiores do que o necessário
- Requer mais tempo

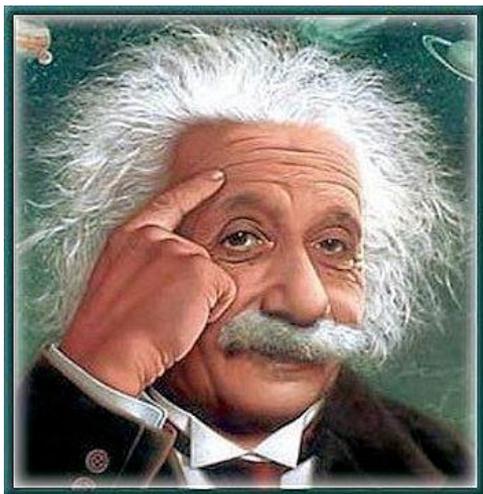
Problemática dos Estimadores



- Desempenho do Estimador
 - Impacta no tempo total de identificação
 - Precisa ser o mais **acurado** possível para **minimizar o número total de *slots*** usados

Problemática dos Estimadores

- Como **estimar de forma mais acurada** possível a quantidade de etiquetas competindo por *slots* em um quadro?



Estimador Eom-Lee



- Propõe um **algoritmo iterativo** para estimar quantidade de etiquetas e tamanho do quadro

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

Valor para k=1 é infinito

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

Valor para k=1 é -2

- **L:** tamanho do quadro
- **Ss:** quantidade de slots bem sucedidos
- **Sc:** quantidade de slots em colisão

Estimador Eom-Lee



Quando $|\gamma_{k^*-1} - \gamma_{k^*}|$
é menor do que um limiar,
o processo termina.

$$\hat{f} = \gamma_{k^*} \cdot s_c .$$

Tamanho do Próximo Quadro

$$\hat{n} = \frac{\hat{f}}{\beta_{k^*}} .$$

Estimativa de etiquetas que competiram por *slots*

E o desempenho ...?



Eom-Lee é um bom estimador?

... Vamos ver mais na frente!





Estimador Vogt

- Utiliza conceitos de probabilidade para a estimativa
- Assume uma **distribuição binomial**
- A quantidade de *slots* contendo transmissões de **r etiquetas** em um **quadro de tamanho L** é:

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

- **n**: população de etiquetas que competem por *slots*



Estimador Vogt

- Com base na Eq. Anterior, Vogt define ...

$$\epsilon(L, s_v, s_s, s_c, n) = \left\| \begin{pmatrix} a_0^{L,n} \\ a_1^{L,n} \\ a_{\geq 2}^{L,n} \end{pmatrix} - \begin{pmatrix} s_v \\ s_s \\ s_c \end{pmatrix} \right\|$$

- $\|\bullet\|$ representa a norma Euclidiana

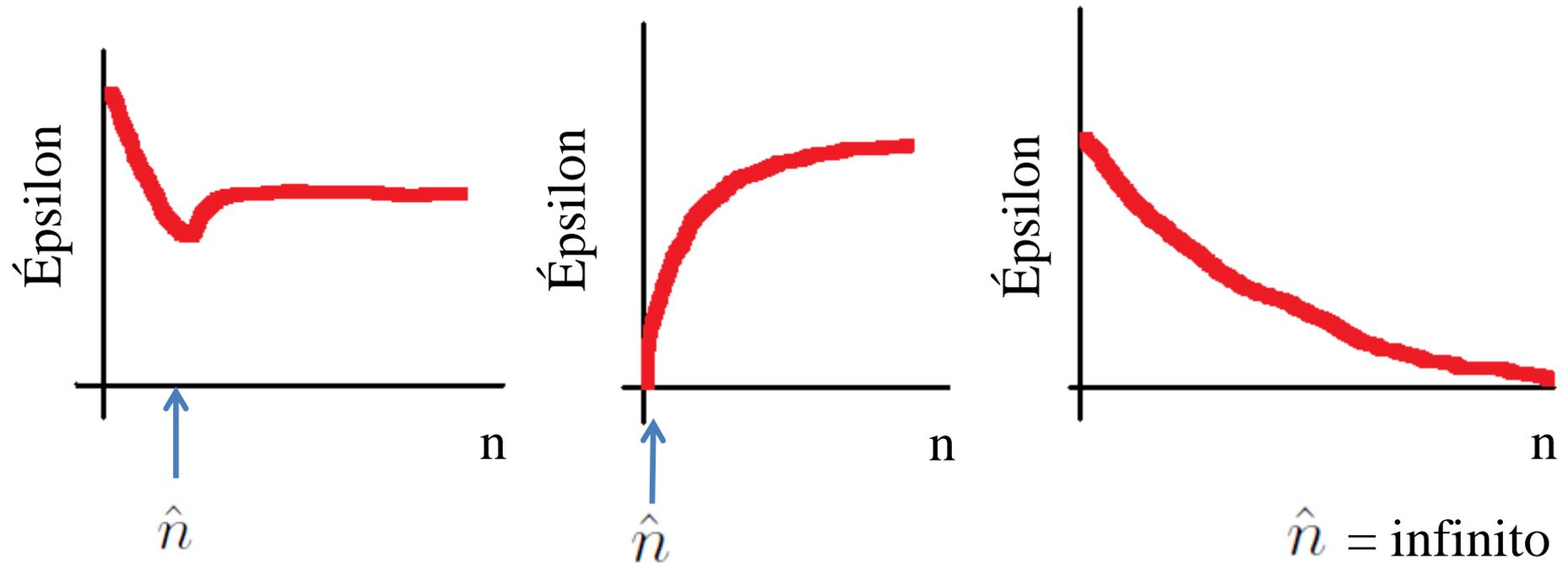
$$\hat{n} = \operatorname{argmin}_{n \geq 1} \epsilon(L, s_v, s_s, s_c, n)$$

Estimativa de etiquetas que competiram por *slots*

Estimador Vogt



$$\hat{n} = \operatorname{argmin}_{n \geq 1} \epsilon(L, s_v, s_s, s_c, n)$$



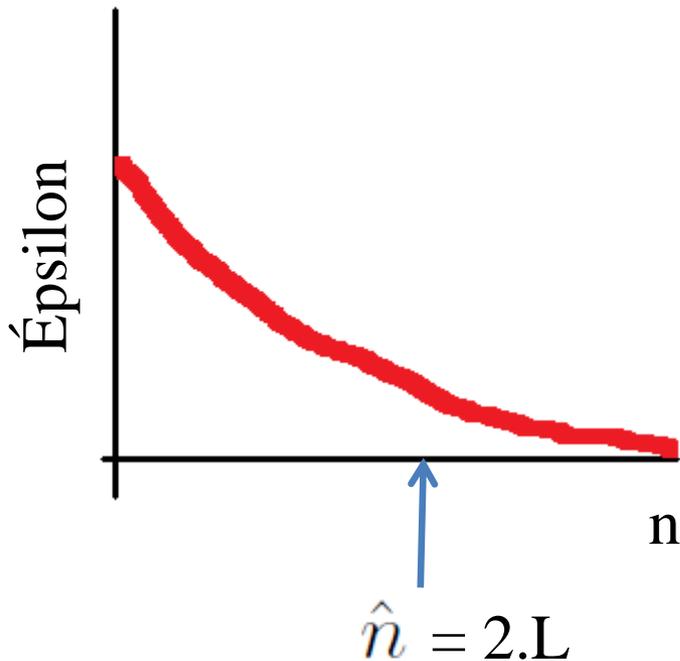
- **L**: tamanho do quadro
- **n**: população de etiquetas
- **Sv**: quantidade de slots vazios

- **Ss**: quantidade de slots bem sucedidos
- **Sc**: quantidade de slots em colisão
- \hat{n} : estimativa de etiquetas

Estimador Vogt



$$\hat{n} = \operatorname{argmin}_{n \geq 1} \epsilon(L, s_v, s_s, s_c, n)$$



- **Solução do Vogt?** Usar o Lower Bound !!!

- **L:** tamanho do quadro
- **n:** população de etiquetas
- **Sv:** quantidade de slots vazios

- **Ss:** quantidade de slots bem sucedidos
- **Sc:** quantidade de slots em colisão
- \hat{n} : estimativa de etiquetas

E o desempenho ...?



Vogt e Eom-Lee são bons estimadores?

... Vamos ver mais na frente!



Mas segundo Eom-Lee ...



- **Comparação** com outros estimadores
 - Lower Bound, Schoute, Vogt (Eom-Lee), Chen, C-Ratio
 - Quadro inicial de 64 *slots*
 - População desconhecida de 100 a 1000 etiquetas
- Eom-Lee **possui a melhor acurácia**
 - exceto quando o tamanho do quadro é próximo do tamanho da população de etiquetas
- Eom-Lee conclui o processo de identificação com **quantidade menor ou similar de *slots***

Objetivo do Trabalho



- Vamos partir do Vogt ...
- Há como obter uma **melhor estimativa** para o caso de todos os *slots* do quadro estarem em colisão?
- Vamos **propor duas abordagens** cada qual em um estimador
 - Improved Vogt I (**IV-I**)
 - Improved Vogt II (**IV-II**)
- Qual o **impacto** no processo de identificação?

O Estimador IV-I Proposto



Algoritmo 4.1 Estimador IV-I proposto. Entradas δ , L , s_s , s_v e s_c .

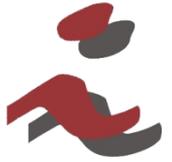
```
1: Se ( $L \neq s_c$ ) então  
2:    $\hat{n} \leftarrow$  Resultado da Eq. (3);  
3: Senão  
4:    $n \leftarrow 2 \times L$ ;  
5:    $a_0 \leftarrow L \times \left(1 - \frac{1}{L}\right)^n$ ;  
6:    $a_1 \leftarrow n \times \left(1 - \frac{1}{L}\right)^{n-1}$ ;  
7:    $\epsilon \leftarrow \sqrt{2} \times \sqrt{a_0^2 + a_1^2 + (a_0 \times a_1)}$ ;  
8:    $\epsilon \leftarrow$  Truncar( $\delta \times \epsilon$ );  
9:    $\epsilon_{anterior} \leftarrow \epsilon + 1$ ;  
10:  Enquanto ( $\epsilon < \epsilon_{anterior}$ ) faça  
11:     $n \leftarrow n + 1$ ;  
12:     $\epsilon_{anterior} \leftarrow \epsilon$ ;  
13:     $a_0 \leftarrow L \times \left(1 - \frac{1}{L}\right)^n$ ;  
14:     $a_1 \leftarrow n \times \left(1 - \frac{1}{L}\right)^{n-1}$ ;  
15:     $\epsilon \leftarrow \sqrt{2} \times \sqrt{a_0^2 + a_1^2 + (a_0 \times a_1)}$ ;  
16:     $\epsilon \leftarrow$  Truncar( $\delta \times \epsilon$ );  
17:  fim Enquanto  
18:   $\hat{n} \leftarrow n - 1$ ;  
19: fim Senão  
20: Retorne( $\hat{n}$ );
```

Como no Vogt

Inicializa

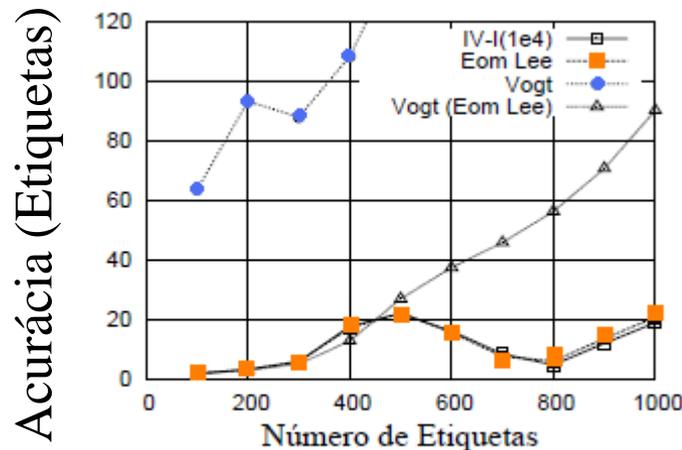
Percorre a
exponencial

Acurácia do IV-I



Quadro Inicial de 64 slots

Etiquetas	Vogt	Vogt (Eom-Lee)	Eom-Lee	IV-I ($\delta = 1e4$)
100	63,59 \pm 0,78	(*) 1,82 \pm 0,04	1,92 \pm 0,04	1,95 \pm 0,05
200	93,22 \pm 0,89	(*) 3,17 \pm 0,08	(*) 3,25 \pm 0,08	3,28 \pm 0,09
300	87,92 \pm 0,86	(*) 5,07 \pm 0,18	6,02 \pm 0,43	5,84 \pm 0,40
400	108,36 \pm 0,78	(*) 12,91 \pm 0,56	17,21 \pm 0,89	17,78 \pm 0,89
500	143,08 \pm 0,87	26,98 \pm 0,53	(*) 22,21 \pm 0,33	22,11 \pm 0,37
600	185,33 \pm 0,75	37,46 \pm 0,42	(*) 15,68 \pm 0,12	16,15 \pm 0,12
700	233,12 \pm 0,68	45,84 \pm 0,48	(*) 7,69 \pm 0,16	8,49 \pm 0,13
800	286,68 \pm 0,71	56,30 \pm 0,75	(*) 6,16 \pm 0,08	4,85 \pm 0,08
900	351,87 \pm 0,72	70,68 \pm 1,13	(*) 13,60 \pm 0,09	12,07 \pm 0,09
1000	425,24 \pm 0,70	90,15 \pm 1,19	(*) 20,80 \pm 0,13	18,97 \pm 0,13



5 em 10

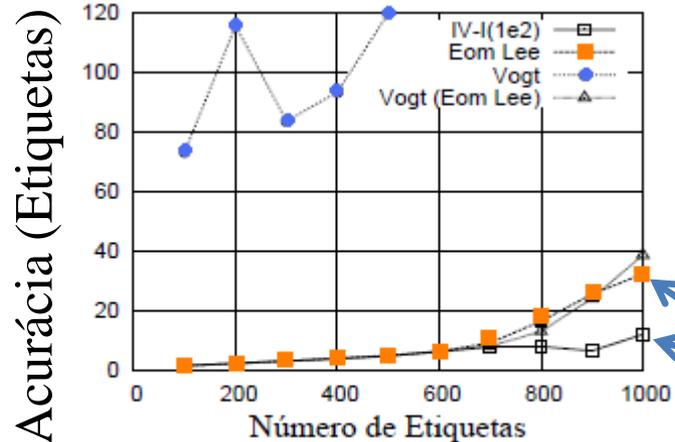


Acurácia do IV-I



Quadro Inicial de 128 slots

Etiquetas	Vogt	Vogt (Eom-Lee)	Eom-Lee	IV-I ($\delta = 1e2$)
100	73,23 \pm 0,86	(^f) 1,54 \pm 0,03	(^f) 1,53 \pm 0,03	1,62 \pm 0,04
200	115,52 \pm 0,81	(^f) 2,26 \pm 0,04	2,38 \pm 0,04	2,35 \pm 0,05
300	83,61 \pm 0,53	(^f) 3,11 \pm 0,07	(^f) 3,19 \pm 0,06	3,12 \pm 0,06
400	93,34 \pm 0,71	(^f) 3,92 \pm 0,09	(^f) 4,00 \pm 0,09	3,97 \pm 0,09
500	119,75 \pm 0,51	(^f) 4,76 \pm 0,12	5,04 \pm 0,13	4,87 \pm 0,13
600	155,97 \pm 0,66	(^f) 6,35 \pm 0,21	(^f) 6,37 \pm 0,20	6,27 \pm 0,19
700	203,71 \pm 0,64	(^f) 8,14 \pm 0,35	9,19 \pm 0,59	7,90 \pm 0,25
800	260,02 \pm 0,72	(*) 13,04 \pm 0,73	16,52 \pm 1,14	8,00 \pm 0,17
900	328,42 \pm 0,72	(*) 24,15 \pm 1,16	(*) 25,54 \pm 1,28	6,55 \pm 0,21
1000	404,84 \pm 0,73	38,66 \pm 1,19	(*) 32,32 \pm 0,84	12,07 \pm 0,15



9 em 10

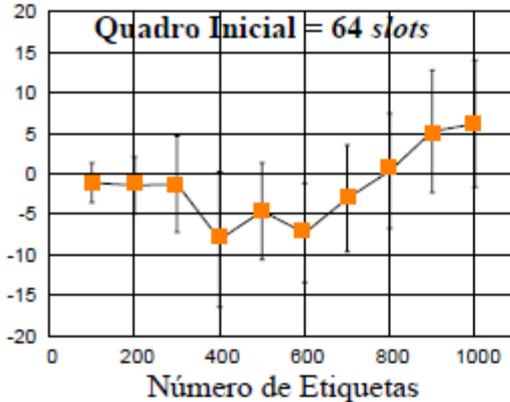
2,67 vezes

Impacto da Acurácia

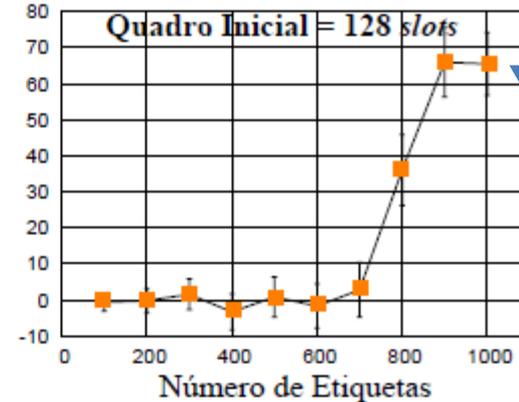


IV-I em relação ao Eom-Lee

Redução (slots)



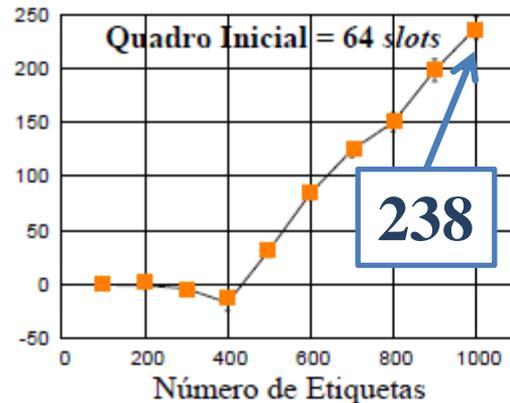
Redução (slots)



66

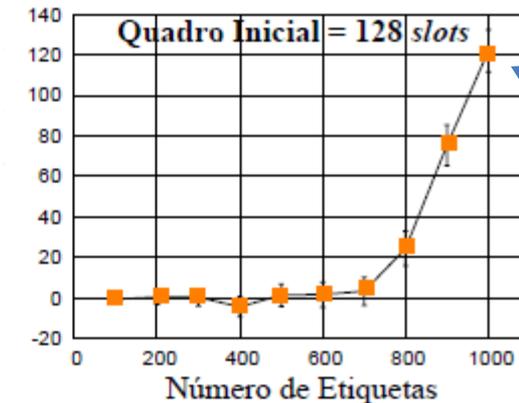
IV-I em relação ao Vogt (Eom-Lee)

Redução (slots)



238

Redução (slots)



121

Melhorando o IV-I



Algoritmo 4.1 Estimador IV-I proposto. Entradas δ , L , s_s , s_v e s_c .

```
1: Se ( $L \neq s_c$ ) então  
2:    $\hat{n} \leftarrow$  Resultado da Eq. (3);  
3: Senão  
4:    $n \leftarrow 2 \times L$ ;  
5:    $a_0 \leftarrow L \times \left(1 - \frac{1}{L}\right)^n$ ;  
6:    $a_1 \leftarrow n \times \left(1 - \frac{1}{L}\right)^{n-1}$ ;  
7:    $\epsilon \leftarrow \sqrt{2} \times \sqrt{a_0^2 + a_1^2 + (a_0 \times a_1)}$ ;  
8:    $\epsilon \leftarrow$  Truncar( $\delta \times \epsilon$ );  
9:    $\epsilon_{anterior} \leftarrow \epsilon + 1$ ;  
10:  Enquanto ( $\epsilon < \epsilon_{anterior}$ ) faça  
11:     $n \leftarrow n + 1$ ;  
12:     $\epsilon_{anterior} \leftarrow \epsilon$ ;  
13:     $a_0 \leftarrow L \times \left(1 - \frac{1}{L}\right)^n$ ;  
14:     $a_1 \leftarrow n \times \left(1 - \frac{1}{L}\right)^{n-1}$ ;  
15:     $\epsilon \leftarrow \sqrt{2} \times \sqrt{a_0^2 + a_1^2 + (a_0 \times a_1)}$ ;  
16:     $\epsilon \leftarrow$  Truncar( $\delta \times \epsilon$ );  
17:  fim Enquanto  
18:   $\hat{n} \leftarrow n - 1$ ;  
19: fim Senão  
20: Retorne( $\hat{n}$ );
```

Como no Vogt

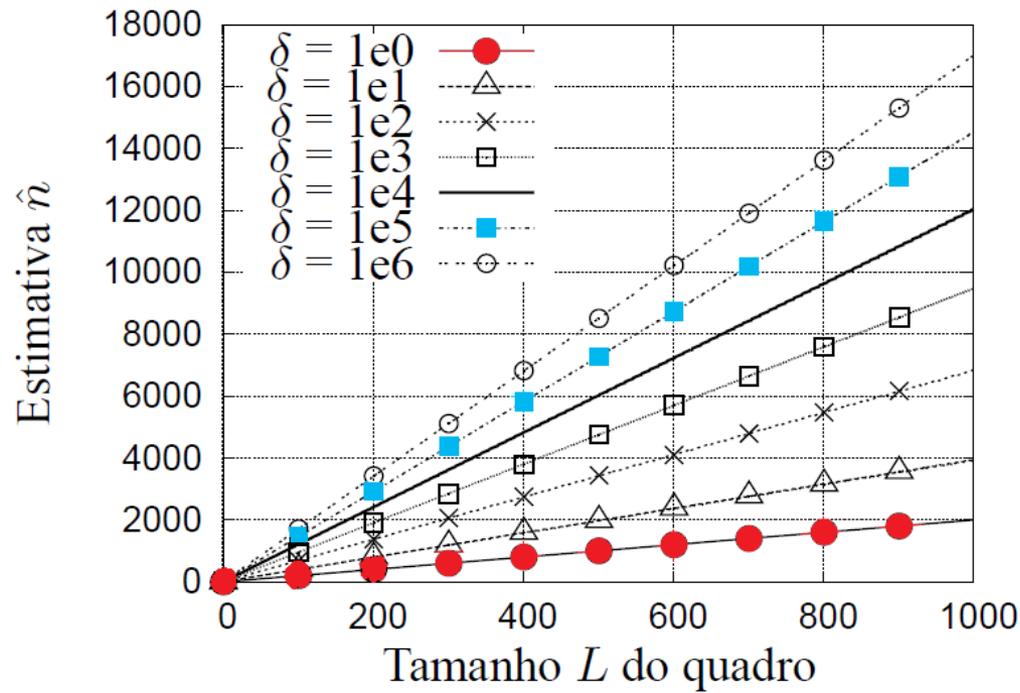
Inicializa

Percorre a
exponencial

Melhorando o IV-I



- Proposta
 - Para o caso em que todos os *slots* estejam em colisão, aproximar a função de estimativa da população de etiquetas por uma reta



O Estimador IV-II Proposto



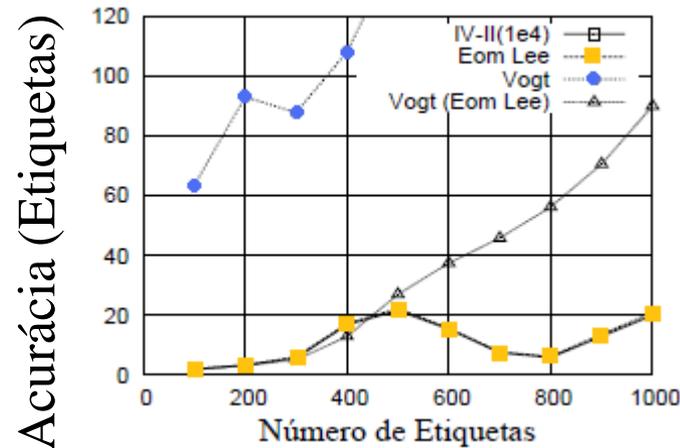
Algoritmo 5.1 Estimador IV-II proposto. Entradas δ , L , s_s , s_v e s_c .

```
1: Se ( $L \neq s_c$ ) então  
2:    $\hat{n} \leftarrow$  Resultado da Eq. (3);  
3: Senão  
4:   Comute para ( $\delta$ )  
5:     1e0 :  $n \leftarrow 2,001001000 * (L - 1) + 2$ ; interrompa;  
6:     1e1 :  $n \leftarrow 3,947947950 * (L - 1) + 2$ ; interrompa;  
7:     1e2 :  $n \leftarrow 6,851851850 * (L - 1) + 2$ ; interrompa;  
8:     1e3 :  $n \leftarrow 9,497497500 * (L - 1) + 2$ ; interrompa;  
9:     1e4 :  $n \leftarrow 12,047047047 * (L - 1) + 2$ ; interrompa;  
10:    1e5 :  $n \leftarrow 14,518518500 * (L - 1) + 2$ ; interrompa;  
11:    1e6 :  $n \leftarrow 17,011011000 * (L - 1) + 2$ ; interrompa;  
12:   fim Comute  
13:    $\hat{n} \leftarrow \lceil n \rceil$ ;  
14: fim Senão  
15: Retorne ( $\hat{n}$ );
```

Como no Vogt

Aproximações

Acurácia do IV-II



7 em 10

Quadro Inicial de 64 slots

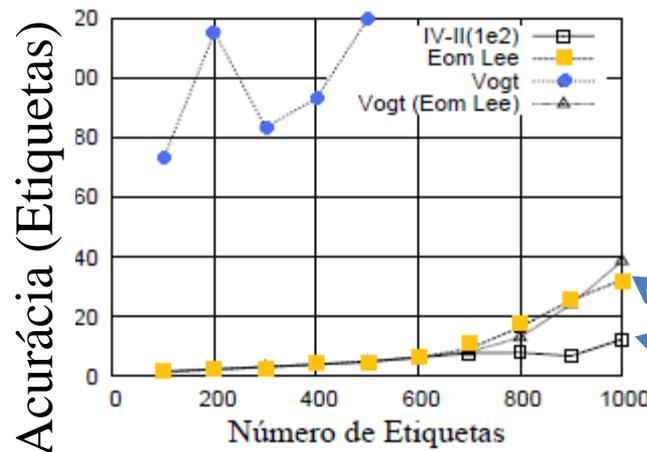
Etiquetas	Vogt	Vogt (Eom-Lee)	Eom-Lee	IV-II ($\delta = 1e4$)
100	63,59 ± 0,78	1,82 ± 0,04	1,92 ± 0,04	1,91 ± 0,04
200	93,22 ± 0,89	3,17 ± 0,08	3,25 ± 0,08	3,25 ± 0,08
300	87,92 ± 0,86	5,07 ± 0,18	6,02 ± 0,43	5,68 ± 0,36
400	108,36 ± 0,78	12,91 ± 0,56	17,21 ± 0,89	16,82 ± 0,85
500	143,08 ± 0,87	26,98 ± 0,53	22,21 ± 0,33	21,59 ± 0,34
600	185,33 ± 0,75	37,46 ± 0,42	15,68 ± 0,12	15,37 ± 0,13
700	233,12 ± 0,68	45,84 ± 0,48	7,69 ± 0,16	7,48 ± 0,12
800	286,68 ± 0,71	56,30 ± 0,75	6,16 ± 0,08	5,87 ± 0,09
900	351,87 ± 0,72	70,68 ± 1,13	13,60 ± 0,09	12,96 ± 0,10
1000	425,24 ± 0,70	90,15 ± 1,19	20,80 ± 0,13	19,98 ± 0,15

Acurácia do IV-II



Quadro Inicial de 128 slots

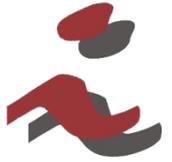
Etiquetas	Vogt	Vogt (Eom-Lee)	Eom-Lee	IV-II ($\delta = 1e2$)
100	73,23 \pm 0,86	1,54 \pm 0,03	1,53 \pm 0,03	1,57 \pm 0,03
200	115,52 \pm 0,81	2,26 \pm 0,04	2,38 \pm 0,04	2,31 \pm 0,05
300	83,61 \pm 0,53	3,11 \pm 0,07	3,19 \pm 0,06	3,11 \pm 0,06
400	93,34 \pm 0,71	3,92 \pm 0,09	4,00 \pm 0,09	3,96 \pm 0,09
500	119,75 \pm 0,51	4,76 \pm 0,12	5,04 \pm 0,13	4,91 \pm 0,13
600	155,97 \pm 0,66	6,35 \pm 0,21	6,37 \pm 0,20	6,28 \pm 0,20
700	203,71 \pm 0,64	8,14 \pm 0,35	9,19 \pm 0,59	7,75 \pm 0,25
800	260,02 \pm 0,72	13,04 \pm 0,73	16,52 \pm 1,14	7,96 \pm 0,17
900	328,42 \pm 0,72	24,15 \pm 1,16	25,54 \pm 1,28	6,80 \pm 0,21
1000	404,84 \pm 0,73	38,66 \pm 1,19	32,32 \pm 0,84	12,36 \pm 0,16



10 em 10

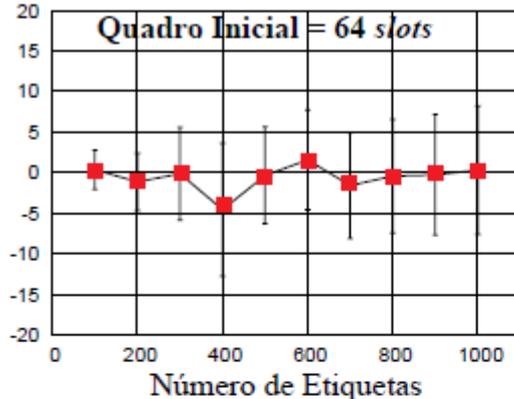
2,61 vezes

Impacto da Acurácia

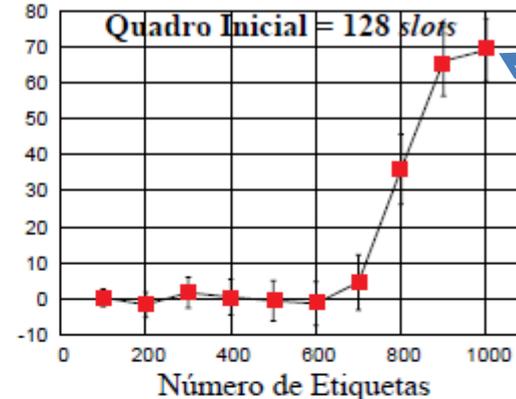


IV-II em relação ao Eom-Lee

Redução (slots)



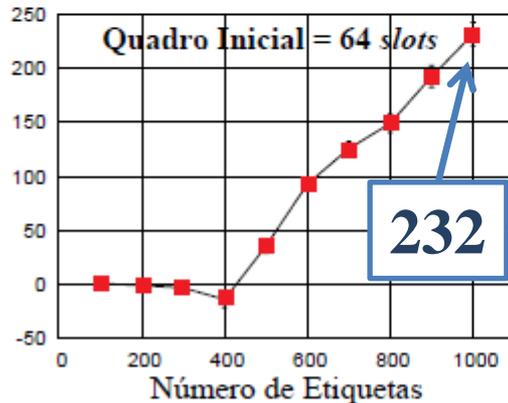
Redução (slots)



69

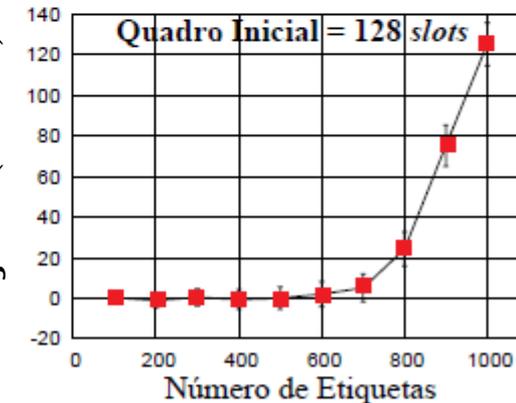
IV-II em relação ao Vogt (Eom-Lee)

Redução (slots)



232

Redução (slots)



125

Conclusões



- Proposta de 2 estimadores para o DFSA
 - IV-I e IV-II
- IV-II é melhor do que o IV-I no confronto direto com o Vogt, o Vogt (Eom-Lee) e o Eom-Lee
- Quadro inicial de 64 slots e população desconhecida entre 100 e 1000 etiquetas
 - Eom-Lee e IV-II usam quantidade equivalente de slots
 - IV-II usa quantidade equivalente ou menor de slots do que o Vogt(Eom-Lee)
 - 232 slots a menos quando menor

Conclusões



- Quadro inicial de **128 slots** e população desconhecida entre **100 e 1000 etiquetas**
 - IV-II é **significativamente mais acurado** do que o Vogt
 - IV-II permite quantidade equivalente ou menor de *slots* do que o Eom-Lee e o Vogt (Eom-Lee)
 - **até 69 slots a menos** do que o Eom-Lee
 - **até 125 slots a menos** do que o Vogt (Eom-Lee)

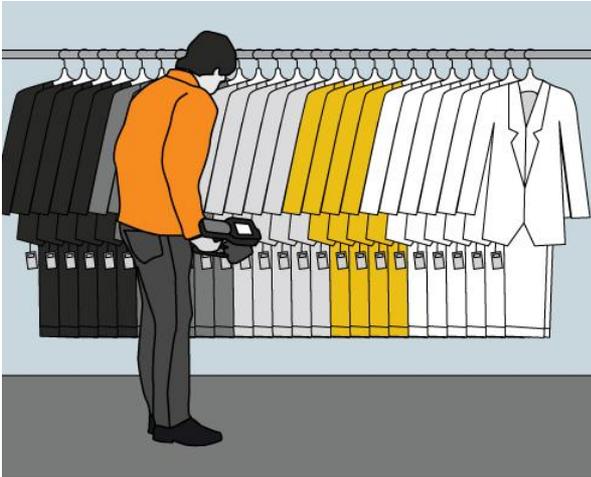


DÚVIDAS?

Júlio D. Andrade
Paulo André da S. Gonçalves







Vogt



$$\epsilon(L, s_v, s_s, s_c, n) = \left\| \begin{pmatrix} a_0^{L,n} \\ a_1^{L,n} \\ a_{\geq 2}^{L,n} \end{pmatrix} - \begin{pmatrix} s_v \\ s_s \\ s_c \end{pmatrix} \right\|$$

$$\hat{n} = \underset{n \geq 1}{\operatorname{argmin}} \quad \epsilon(L, s_v, s_s, s_c, n)$$

- **L**: tamanho do quadro
- **n**: população de etiquetas
- **Sv**: quantidade de slots vazios
- **Ss**: quantidade de slots bem sucedidos
- **Sc**: quantidade de slots em colisão
- \hat{n} : estimativa de etiquetas