



OTIMIZANDO O ALGORITMO-Q PARA SISTEMAS RFID NA INTERNET DAS COISAS

Daniel Carriço de Lima Menezes¹; Paulo André da Silva Gonçalves²

¹Estudante do Curso de Engenharia da Computação- CIn – UFPE; E-mail: dclm@cin.ufpe.br,

²Docente/pesquisador do Depto de Ciência da Computação – CIn –UFPE. E-mail: pasg@cin.ufpe.br.

Sumário: Projeto realizado com o intuito de pesquisar, aprender e otimizar os algoritmos de identificação de etiquetas RFID presentes no mercado, tendo como principal foco o algoritmo Q, cujo padrão EPC Classe 1 Gen2 [EPC Global 2015] o especifica em um de seus procedimentos de identificação de etiquetas. Para tal, teve de ser desenvolvido um simulador, na linguagem C/C++ em ambiente *Linux*, para validar e realizar experimentos de vários ambientes, com diferentes tipos de estimadores, para uma posterior comparação. Sendo finalizado então com a pesquisa e otimização do algoritmo-Q proposto. Tendo como proposta a divisão do fator de correção C em dois fatores distintos CqE e CqC de valoração única independente da quantidade de etiquetas a serem identificadas, 0.1 e 0.4 respectivamente.

Palavras-chave: IoT; RFID; anticolisao;

INTRODUÇÃO

Dentre os protocolos anticolisão para sistemas RFID, o DFSA (Dynamic Framed Slotted ALOHA) é um dos mais populares [EPC Global 2015, Klair et al. 2010]. A execução do DFSA é orientada pelo leitor que organiza em um ou mais quadros. Cada quadro é ainda subdividido em slots de tempo. As etiquetas são requisitadas a transmitir em um slot, aleatoriamente escolhido a cada quadro, até serem identificadas pelo leitor. Contudo, o DFSA requer o uso de alguma técnica para reajuste dinâmico do tamanho dos quadros. Existem vários estudos que fazem isso com base na estimativa da população de etiquetas competindo por slots [Andrade and Gonçalves 2013, Klair et al. 2010] ou com base em algum algoritmo [EPC Global 2015]. Os estimadores ou algoritmos de ajuste de tamanho de quadro para o DFSA em sistemas RFID podem ser divididos em duas classes básicas: com e sem reset de quadro. A primeira classe recalcula o valor do próximo quadro a ser aberto somente ao término da execução de um quadro completo, como é o caso dos algoritmos de [Andrade and Gonçalves 2013, Eom and Lee 2010, Chen 2009, Vogt 2002, Schoute 1983]. A segunda classe decide ao término de cada slot ou de um subgrupo de slots consecutivos se cancela ou não a execução dos slots restantes e recalcula o tamanho do próximo quadro [Chen 2014, EPC Global 2015].

MATERIAIS E MÉTODOS

Utilizou-se um computador *AVELL* com placa gráfica *NVIDIA GTX 1070*, processador *intel(R) core™ i7- 6700HQ*, sistema operacional *linux* 32 GB de RAM. Foi implementado um simulador na linguagem C/C++, sendo feitos primeiro os estimadores de parâmetros citados por [Eom and Lee 2010, Chen 2009, Vogt 2002, Schoute 1983], logo após, foi implementado o algoritmo-Q pelo padrão EPC Classe 1 Gen2 [EPC Global 2015] com quadros ajustados dinamicamente com tamanho 2^Q , com Q recalculado ao término de cada

slot com base no algoritmo-Q. Este procedimento resolve ineficiências típicas de um sistema FSA e nada mais é do que um DFSA com reset de quadro ao término de cada slot vazio ou em colisão. Quando um slot termina, o leitor analisa o tipo de slot. De acordo com o resultado da análise, o leitor atualiza o valor do parâmetro Q_{fp} que nada mais é do que a versão em ponto flutuante do parâmetro inteiro Q e é calculado como segue: $Q_{fp} = \min(15, Q_{fp} + C)$ para slot em colisão, $Q_{fp} = \max(0, Q_{fp} - C)$ para slot vazio e Q_{fp} permanece inalterado para cada slot bem sucedido. O novo valor do parâmetro Q passa a ser o inteiro mais próximo a Q_{fp} . Valores para C são dinâmicos e estão no intervalo de 0,1 a 0,5. A dificuldade está em se determinar valores adequados para C visto que valores ótimos são dependentes do número de etiquetas competindo por slots. O padrão não define como calcular valores adequados para C .

RESULTADOS

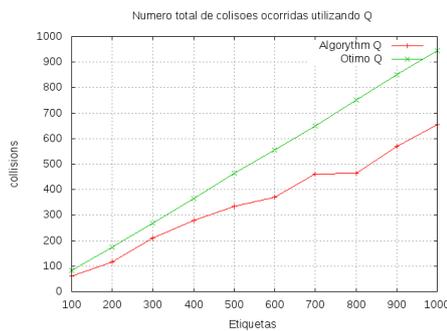


Figura 1. Total de Colisões x # Etiquetas

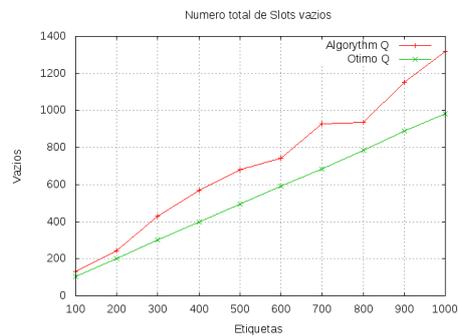


Figura 2. Total de Slots Vazios x # Etiquetas

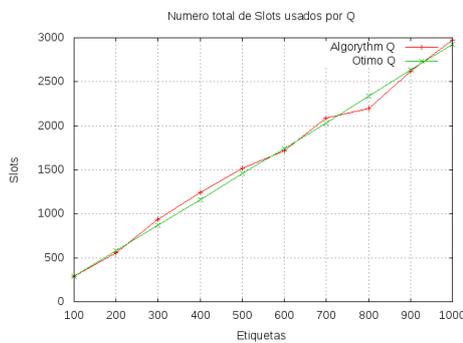


Figura 3. Total de Slots x # Etiquetas

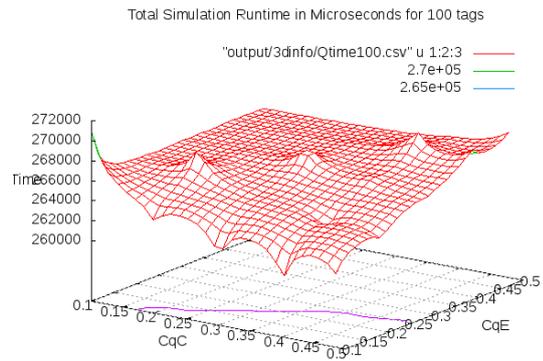


Figura 4. Tempo de Id. x CqC x CqE (100 etiquetas)

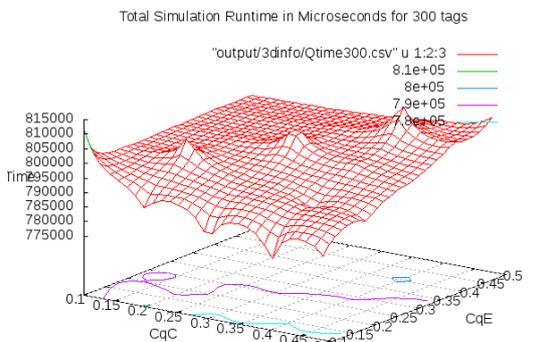


Figura 5. Tempo de Id. x CqC x CqE (300 etiquetas)

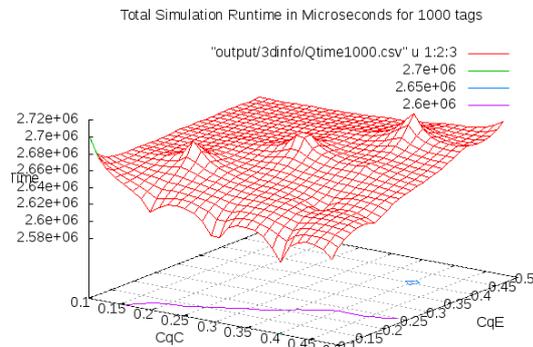


Figura 6. Tempo de Id. x CqC x CqE (1000 etiquetas)

DISCUSSÃO

Resultante da implementação do algoritmo-Q em duas versões, uma seguindo o padrão EPC Classe 1 Gen2 [EPC Global 2015], e outra utilizando de um modo de previsão de etiquetas, onde o sistema já sabe quantas etiquetas têm de identificar, para então já obter os valores ótimos de Q, podemos identificar que o algoritmo ótimo proposto tem mais colisões que o algoritmo-Q original e menos slots vazios (Figuras 1, 2 e 3). Com base nos resultados obtidos até então, concluímos que para fins de mercado, uma otimização no algoritmo não depende somente de menos quantidade de slots, como também um menor tempo de identificação. Sendo então calculados em 3 tipos de tempo diferente, esses tempos são os seguintes conforme norma ISO 18000-6 Type C: 1) tempo de slot bem sucedido = $TS = T4 + TQuery + 2T1 + 2T2 + TRN16 + TACK + TPC + EPC + CRC16 + TQREP = 2312 \mu s$; 2) tempo de slot em colisão $TC = T1 + T2 + TRN16 = 337,5 \mu s$ e 3) tempo de slot vazio = $TE = T1 + T3 = 67,5 \mu s$. Com isso, sabemos que para uma melhoria no algoritmo, seria necessária uma forma de diminuir a quantidade de colisões, podendo aumentar a quantidade de slots vazios até um certo limite, para então uma melhoria no tempo total de identificação de etiquetas. Sendo discutida a possibilidade de manipular a constante C não como um parâmetro único para todos os casos que podem ocorrer ao identificar uma etiqueta, mas como dois parâmetros distintos em que um deve ser utilizado para casos em que ocorrem colisões e outro em que ocorrem em casos de slots vazios. A constante C foi então dividida em duas constantes diferentes, CqC , para casos de *slots* em colisão, e CqE , para casos de *slots* vazios (Figuras 4, 5 e 6).

CONCLUSÕES

Com base nos resultados obtidos, podemos verificar que a divisão do fator de correção C em dois fatores distintos CqC e CqE , apresentam melhores resultados de tempo. Possuindo estes os valores de 0,4 e 0,1 respectivamente, para um melhor resultado de tempo independente da quantidade de etiquetas a ser identificada. Podemos ver também que no algoritmo-Q original, os valores de C ocupam os pontos de maior tempo de identificação, sendo então a versão original do Q, como os piores casos do novo algoritmo-Q proposto.

AGRADECIMENTOS

Em especial, agradeço ao professor Paulo André da Silva Gonçalves por toda sua paciência e tempo disponibilizado para esclarecimento de dúvidas. Ao meu colega e amigo Lucas Augusto Mota de Alcântara, por ter me incentivado e estar disposto a me ajudar.

REFERÊNCIAS

- Perera , C. Liu , C. H. , and Jayawardena , S. (2015). The Emerging Internet of Things Marketplace From an Industrial Perspective: A Survey . IEEE Transactions on Emerging Topics in Computing , 3(4):585-598.
- Andrade, J. D. and Gonçalves, P. A. S. (2013). Um Estimador Acurado para o Protocolo DFSA em Sistemas RFID. In Proc. of XXXI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC), pages 325–338, Brasília, DF.



Chen, W.-T. (2014). A Feasible and Easy-to-Implement Anticollision Algorithm for the EPCglobal UHF Class-1 Generation-2 RFID protocol. *IEEE Transactions on Automation Science and Engineering*, 11:485 – 491.

Eom, J.-B. and Lee, T.-J. (2010). Accurate Tag Estimation for Dynamic Framed-slotted ALOHA in RFID Systems. *IEEE Communications Letters*, 14:60–62.

EPC Global, I. (2015). EPC radio-frequency identify protocols Generation-2 UHF RFID Specification for RFID air interface protocol for communications at 860MHz-960 MHz - ratified, 2.0.1 edition.

Vogt, H. (2002). Efficient Object Identification with Passive RFID Tags. In *Proc. of the First International Conf. on Pervasive Computing*, pages 98–113, London, UK.

Klair, D., Chin, K.-W., and Raad, R. (2010). A Survey and Tutorial of RFID AntiCollision Protocols. *IEEE Communications Surveys Tutorials*, 12(3):400 –421.

Schoute, F. C. (1983). Dynamic Frame Length ALOHA. *IEEE Transactions on Communications*, 31:565–568.