

PHÂN TÍCH CÁC KỸ THUẬT ƯỚC TÍNH THẺ RFID TRONG GIAO THỨC DFSA

Nguyễn Văn Vũ*, Huỳnh Bửu Ngọc

Trường Đại học An Giang

*Email: nvvu@agu.edu.vn

Ngày nhận bài: 27/8/2018; ngày hoàn thành phản biện: 7/9/2018; ngày duyệt đăng: 10/12/2018

TÓM TẮT

RFID là một công nghệ nhận dạng tự động được triển khai trong khá nhiều ứng dụng thực tế ngày nay. Trong một hệ thống RFID, các đầu đọc thực hiện nhận dạng các thẻ thông qua sóng vô tuyến mà không cần tiếp xúc. Một vấn đề kỹ thuật trong nhận dạng thẻ RFID là xung đột thẻ sẽ xảy ra khi có nhiều thẻ được đọc đồng thời bởi một đầu đọc. Một số kỹ thuật ước tính thẻ do đó đã được đề xuất nhằm giảm xung đột, giảm lãng phí băng thông đường truyền, giảm số lần truy vấn và kết quả là tăng hiệu quả đọc thẻ. Bài viết này sẽ phân tích một số kỹ thuật ước tính thẻ đã được đề xuất trong giao thức DFSA và so sánh dựa trên kết quả mô phỏng.

Từ khoá: Hệ thống RFID, DFSA, xung đột thẻ, kỹ thuật ước tính, hiệu quả hệ thống.

1. MỞ ĐẦU

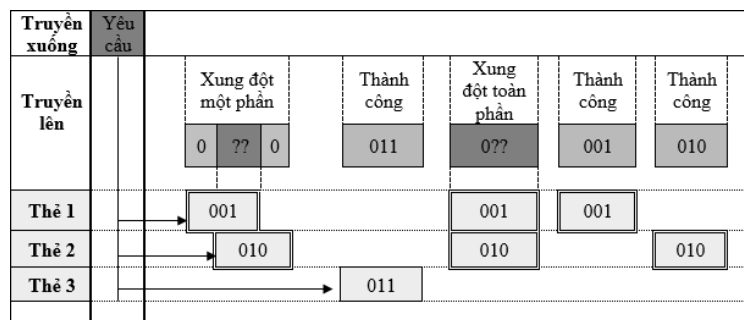
Công nghệ nhận dạng đối tượng bằng sóng vô tuyến (*Radio Frequency Identification - RFID*) được xem là một giải pháp hiệu quả nhằm thay thế công nghệ mã vạch (*barcode*) trong các lĩnh vực như quản lý chuỗi cung ứng, phân phối, nông nghiệp, quân sự, y tế, ngân hàng... Một đầu đọc RFID có thể đọc dữ liệu từ các thẻ RFID mà không cần tiếp xúc và có khả năng đọc nhiều thẻ cùng lúc trong vùng phủ sóng của nó, trong khi các đầu đọc mã vạch chỉ có thể đọc được một mã vạch tại một thời điểm [1].

Một hệ thống RFID bao gồm 3 thành phần: các thẻ RFID (*tag*), các đầu đọc RFID (*reader*) và thành phần trung gian thường là một máy chủ để lưu trữ và quản lý dữ liệu. Thẻ RFID có một ăng-ten để gửi và nhận dữ liệu. Các thẻ RFID được phân làm ba loại chính: thẻ chủ động (*active*), thẻ thụ động (*passive*) và thẻ bán thụ động (*semi-active*,

semi-passive). Đầu đọc RFID có thể ghi dữ liệu vào thẻ và đọc dữ liệu từ thẻ và gửi đến máy chủ để xử lý [2].

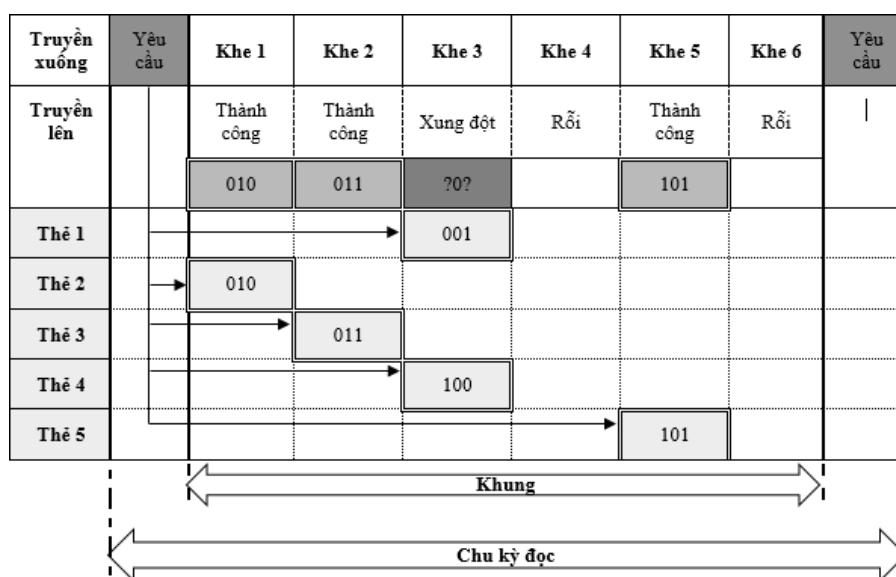
Một vấn đề quan trọng của hệ thống RFID là hiệu quả nhận dạng thẻ. Khi có nhiều thẻ chiếm cùng một kênh truyền thông (*radio frequency - RF*) tại một thời điểm, xung đột (*collision*) sẽ xảy ra; khi đó đầu đọc không thể nhận dạng được dữ liệu của thẻ và kết quả là hiệu quả nhận dạng thẻ thấp. Một giải pháp cho vấn đề này là mở rộng băng tần để tăng tốc độ truyền dữ liệu và giảm thiểu xung đột. Tuy nhiên, việc mở rộng băng tần là không khả thi vì các dải tần số có thể sử dụng là có giới hạn, nên một số kỹ thuật giảm thiểu xung đột khác được đề xuất trong đó có kỹ thuật ước tính thẻ [3].

ALOHA [4] là một giao thức đa truy cập phân chia theo thời gian (*Time Division Multiple Access - TDMA*). Với giao thức này, một thẻ sẽ bắt đầu truyền ngay khi nó sẵn sàng và có dữ liệu để gửi. Chu kỳ đọc là khoảng thời gian giữa hai yêu cầu (*request commands*) liên tiếp và có thể được lặp lại cho đến khi tất cả các thẻ trong vùng phủ sóng đều đã được nhận dạng. Ưu điểm của giao thức này là đơn giản, nhưng nhược điểm là tạo ra nhiều xung đột (một phần và toàn phần), như được thể hiện ở Hình 1.



Hình 1. Ví dụ về xung đột một phần và toàn phần trong giao thức ALOHA

FSA (*Framed Slotted ALOHA*) là một cải tiến của giao thức ALOHA với việc bổ sung khái niệm khung (*frame*) [5]. Mỗi khung được chia thành các khe thời gian (*slots*) bằng nhau. Kích thước khung được xác định bởi đầu đọc. Mỗi thẻ sẽ chọn ngẫu nhiên một khe; nếu không có xung đột xảy ra, thẻ sẽ không phải truyền dữ liệu trong chu kỳ kế tiếp. Ưu điểm của FSA là loại bỏ xung đột một phần, nhưng xung đột toàn phần vẫn xảy ra nếu có nhiều thẻ trong vùng đọc của một đầu đọc. Hình 2 mô tả một chu kỳ hoạt động của giao thức FSA.



Hình 2. Mô tả một chu kỳ hoạt động của giao thức FSA

FSA được phân thành hai loại: BFSA (*Basic Framed Slotted ALOHA*) dựa trên việc sử dụng kích thước khung cố định [6] và DFSA (*Dynamic Framed Slotted ALOHA*) dựa trên việc thay đổi kích thước khung [7]. BFSA duy trì kích thước khung cố định trong suốt quá trình hoạt động nên gây ra nhiều xung đột nếu số lượng thẻ cao, trong khi gây ra lãng phí băng thông nếu số lượng thẻ thấp. DFSA khắc phục được nhược điểm này bằng cách thay đổi kích thước khung chuyển biến theo số lượng thẻ: khi số lượng thẻ ít, kích thước khung được điều chỉnh nhỏ; nhưng khi số lượng thẻ cao, kích thước khung được điều chỉnh tăng. Việc điều chỉnh kích thước khung trong DFSA được dựa trên kết quả của chu kỳ trước đó gồm các thông tin: số khe rỗi, số khe xung đột và số khe chỉ có một thẻ. So với BFSA, DFSA có xác suất xung đột thấp hơn; nhưng nếu số lượng thẻ nhiều, xác suất bị xung đột thẻ sẽ cao vì kích thước khung không thể tăng liên tục trong nhiều chu kỳ [8]. Tuy vậy, việc ước tính thẻ để xác định kích thước khung tối ưu luôn là một vấn đề rất quan trọng trong các hệ thống RFID, nhằm làm giảm xung đột và tăng hiệu quả của hệ thống. Bài viết này sẽ phân tích một số kỹ thuật ước tính thẻ đã được đề xuất cho giao thức DFSA, từ đó đề xuất một số hướng phát triển nhằm nâng cao hiệu quả đọc thẻ của nó.

2. VẤN ĐỀ ƯỚC TÍNH THẺ TRONG GIAO THỨC DFSA

Trong giao thức DFSA, việc thay đổi kích thước khung phù hợp được dựa trên kết quả ước tính của chu kỳ trước đó. Gọi E là số khe rỗi, S là số khe chỉ có một thẻ và C là số khe xung đột, ta có một bộ ba $\langle E, S, C \rangle$.

Có thể sử dụng công thức của Schout (1) hoặc Vogt (2) để ước tính số thẻ chưa đọc.

❖ Công thức của Schout [9], [10]:

$$n = S + 2.39C \quad (1)$$

❖ Công thức của Vogt [9], [10]:

Đầu tiên, số thẻ chưa đọc ít nhất được ước tính dựa vào số khe xung đột và số khe thành công. Mỗi khe xung đột có ít nhất là hai thẻ cùng chọn để truyền số ID, do đó số thẻ chưa đọc ít nhất được ước tính bằng Công thức (2).

$$n = S + 2C \quad (2)$$

Tiếp theo, để ước tính số lượng thẻ chưa đọc chính xác hơn, công thức ước tính kết hợp với bộ ba $\langle E, S, C \rangle$ có thể được sử dụng.

$$n_{\text{Vogt}} = \min_n \left| \begin{pmatrix} a_0 \\ a_1 \\ a_k \end{pmatrix} - \begin{pmatrix} E \\ S \\ C \end{pmatrix} \right| \quad (3)$$

trong đó, a_0, a_1, a_k tương ứng với giá trị mong đợi của số khe rỗi, số khe chỉ có một thẻ và số khe xung đột được tính như trong [9], [10].

Khi đầu đọc sử dụng kích thước khung hiện tại F và số lượng thẻ ước tính nhỏ nhất n từ Công thức (1), thì xác suất các thẻ tồn tại trong một khe thời gian r được tính bởi Công thức (4).

$$B_{n, \frac{1}{F}}(r) = \binom{n}{r} \left(\frac{1}{F} \right)^r \left(1 - \frac{1}{F} \right)^{n-r} \quad (4)$$

Kết quả mong đợi của khe thời gian r trong chu kỳ đọc được tính bởi Công thức (5)

$$a_r^{F,n} = F B_{n, \frac{1}{F}}(r) = F \binom{n}{r} \left(\frac{1}{F} \right)^r \left(1 - \frac{1}{F} \right)^{n-r} \quad (5)$$

Từ đó có thể suy ra a_0, a_1, a_k :

$$a_0 = F \left(1 - \frac{1}{F} \right)^n; \quad a_1 = n \left(1 - \frac{1}{F} \right)^{n-1}; \quad a_k = F - a_0 - a_1. \quad (6)$$

Sau khi ước tính được số lượng thẻ chưa đọc, vấn đề đặt ra là thiết lập kích thước khung như thế nào cho phù hợp. DSFA có thể sử dụng bảng tra cứu được đề xuất trong [9] (Bảng 1) để lựa chọn kích thước khung phù hợp cho chu kỳ tiếp theo.

Bảng 1. Bảng ngưỡng trên, ngưỡng dưới và kích thước khung tối ưu tương ứng

Kích thước khung tối ưu	1	4	8	16	32	64	128	256
Ngưỡng dưới	-	-	-	1	10	17	51	112
Ngưỡng trên	-	-	-	9	27	56	129	∞

Khi số thẻ ước tính vượt quá ngưỡng trên, đầu đọc tăng kích thước khung; nhưng khi số thẻ ước tính nhỏ hơn ngưỡng dưới, đầu đọc sẽ giảm kích thước khung. Việc tăng hoặc giảm kích thước khung có thể được thiết lập bằng cách lấy kích thước khung hiện tại nhân hoặc chia cho công bội là 2 [9]. Việc điều chỉnh kích thước khung được lặp lại cho đến khi xác định được kích thước khung phù hợp cho chu kỳ tiếp theo. Quá trình nhận dạng được lặp lại cho đến khi không có xung đột xảy ra, có nghĩa là tất cả các thẻ trong vùng phủ sóng được nhận dạng.

Trong DSFA, do kích thước khung được khởi tạo nhỏ, nên trong trường hợp số lượng thẻ cần nhận dạng lớn thì DFSA phải tăng kích thước khung lên tối đa (256 khe) trong nhiều chu kỳ liên tục; tuy nhiên kích thước khung tối đa này vẫn nhỏ hơn nhiều so với số thẻ chưa nhận dạng và kết quả là xung đột thẻ vẫn cao. Một số cải tiến cho DFSA do đó đã được đề xuất [11], [12] mà sẽ được phân tích kỹ hơn trong phần tiếp theo.

3. PHÂN TÍCH MỘT SỐ KỸ THUẬT ƯỚC TÍNH THẺ TRONG GIAO THỨC DSFA

3.1. Kỹ thuật EDFSA

Với trường hợp số lượng thẻ ước tính được là lớn hơn rất nhiều so với kích thước khung tối đa (256), kỹ thuật EDFSA (*Enhanced Dynamic Framed Slotted ALOHA*) trong [11] có thể được sử dụng. Kỹ thuật này hạn chế số lượng thẻ trả lời đến đầu đọc bằng cách chia số thẻ chưa đọc vào các nhóm và chỉ cho phép một nhóm thẻ trả lời. Số lượng nhóm được tính bằng Công thức (7).

$$M = \left\lceil \frac{\text{Số lượng thẻ chưa đọc}}{F_{\max}} \right\rceil \quad (7)$$

trong đó $F_{\max}$ là kích thước khung tối đa.

Rõ ràng M có thể nhận nhiều giá trị khác nhau. Để đơn giản cho việc cài đặt, EDFSA thiết lập số nhóm là 2, 4, 8, ... Sau khi ước tính số lượng thẻ chưa đọc, EDFSA kết hợp cả việc đặt lại kích thước khung và đồng thời đưa ra số nhóm trong chu kỳ tiếp theo dựa vào kết quả đã ước tính.

Trong các kỹ thuật ước tính thẻ, hiệu quả của hệ thống sẽ giảm khi số lượng thẻ lớn hơn nhiều so với kích thước khung tối đa. Với kích thước khung hiện tại F và số

lượng thẻ ước tính n , kết quả mong đợi của khe thành công trong chu kỳ đọc được viết lại bởi Công thức (8) là

$$a_1^{F,n} = FB_{n, \frac{1}{F}}(1) = n.F \left(\frac{1}{F} \right) \left(1 - \frac{1}{F} \right)^{n-1} \quad (8)$$

trong đó $a_1^{F,n}$ là số khe thời gian mong đợi chỉ có một thẻ đăng ký. Khi đó hiệu quả của hệ thống được tính bằng Công thức (9).

$$SE = \frac{a_1^{F,n}}{F} \quad (9)$$

Khi số lượng thẻ ước tính không lớn hơn nhiều so với kích thước khung tối đa, thì việc chia các thẻ thành hai hay một nhóm với kích thước khung 256 đều cho hiệu quả hệ thống gần bằng nhau. Công thức (10) với trường hợp chia thành hai nhóm có thể suy ra từ Công thức (9).

$$\frac{a_1^{256, n/2}}{256} = \frac{a_1^{256, n}}{256} \quad (10)$$

Thay (8) vào (10), Công thức (11) thu được là

$$\frac{n}{2} \left(\frac{1}{256} \right) \left(1 - \frac{1}{256} \right)^{\frac{n}{2}-1} = n \left(\frac{1}{256} \right) \left(1 - \frac{1}{256} \right)^{n-1} \quad (11)$$

Như vậy việc EDFSA dựa vào số lượng ước tính thẻ để thiết lập đồng thời kích thước khung và số nhóm thì sẽ làm cho thuật toán đơn giản hơn là tách rời hai công việc này. Nếu số lượng thẻ ước tính lớn hơn 354, để đạt hiệu quả hệ thống tối ưu, EDFSA phải chia số lượng thẻ chưa đọc thành hai nhóm. Tương tự, Công thức (11) cũng giúp tính được nếu số lượng thẻ ước tính lớn hơn hoặc bằng 708 thì sẽ thiết lập thành bốn nhóm. Bảng 2 mô tả cách thiết lập kích thước khung và số nhóm tương ứng với số thẻ chưa đọc.

Bảng 2. Số thẻ chưa đọc so với kích thước khung tối ưu và số nhóm

Số thẻ chưa đọc	Kích thước khung (F)	Số nhóm (M)
...	...	...
1417 – 2831	256	8
708 – 1416	256	4
355 – 707	256	2
177 – 354	256	1
82 – 176	128	1

Số thẻ chưa đọc	Kích thước khung (F)	Số nhóm (M)
41 – 81	64	1
20 – 40	32	1
12–19	16	1
6–11	8	1
...	...	...

Một vấn đề có thể nhận thấy rằng, khi số lượng thẻ ước tính nhiều hơn kích thước khung tối đa, EDFSA sử dụng kỹ thuật nhóm thẻ nhằm hạn chế số lượng thẻ trả lời đến đầu đọc, nên số khe xung đột sẽ giảm và tăng hiệu quả hệ thống. Tuy nhiên, điều này cũng làm tăng số khe rỗi do khi chia nhóm thì số thẻ cho phép trả lời đến đầu đọc nhỏ hơn so với kích thước khung tối đa. Kỹ thuật 2CTE (2 *Conditional Tag Estimation*) trong [8] đã khắc phục được nhược điểm này bằng cách bên cạnh việc cố gắng giảm khe xung đột khi kích thước khung nhỏ hơn số lượng thẻ, nó còn cố gắng giảm số lượng khe rỗi khi kích thước khung lớn hơn số lượng thẻ.

3.2. Kỹ thuật 2CTE

2CTE sử dụng kết quả bộ ba $\langle E, S, C \rangle$ của chu kỳ trước để ước tính số thẻ chưa đọc nhỏ nhất từ Công thức (3), đồng thời đặt các giá trị như xác suất trung bình số khe rỗi tối ưu là $OES=0.37$, xác suất trung bình số khe thành công tối ưu $OSS=0.37$ và xác suất trung bình số khe xung đột tối ưu $OCS=0.26$. Các giá trị xác suất này được xác định khi hiệu quả hệ thống tốt nhất đạt được trong giao thức DFSA [4]. Ngoài ra, 2CTE dựa vào xác suất trong khung hiện tại như xác suất số khe rỗi B_0 , xác suất số khe thành công B_1 , và xác suất số khe xung đột B_k để tính độ lệch tương ứng giữa xác suất trung bình tối ưu và xác suất trong khung hiện tại. Các giá trị xác suất B_r được tính bằng Công thức (4).

Để giảm số khe xung đột, 2CTE căn cứ vào xác suất xung đột tồn tại trong mỗi khe thời gian. Nếu xác suất số khe xung đột trong chu kỳ trước đó lớn hơn 26% thì giảm khe xung đột, tức là tăng kích thước khung trong chu kỳ tiếp theo. Để tăng kích thước khung không vượt quá số thẻ chưa đọc, 2CTE xét thêm hai trường hợp sau:

– Nếu xác suất số khe xung đột B_k lớn hơn 50% thì số thẻ chưa đọc (kích thước khung chu kỳ tiếp theo) được ước tính bằng Công thức (12).

$$n' = 2C + (0.75 - (2E_0))(F) \quad (12)$$

– Nếu xác suất số khe xung đột B_k lớn hơn 26% và nhỏ hơn 50% thì ước tính thẻ chưa đọc được tính bằng Công thức (13).

$$n' = 2C + (0.75 - (E_0 + S_0))(F) - S \quad (13)$$

Ngoài ra, 2CTE cũng cố gắng giảm khe rỗi, tức là giảm kích thước khung trong chu kỳ tiếp theo khi số khe thành công và số khe rỗi lớn hơn số khe xung đột và xác của suất số khe xung đột nhỏ hơn 26%. Khi giảm kích thước khung trong chu kỳ tiếp theo, 2CTE chia làm hai trường hợp:

– Nếu xác suất số khe xung đột B_k nhỏ hơn 26% và xác suất số khe rỗi B_0 nhỏ hơn 50% thì ước tính thẻ chưa đọc được tính bằng Công thức (14)

$$n' = n - S \quad (14)$$

– Nếu xác suất số khe xung đột B_k nhỏ hơn 26% và xác suất số khe rỗi B_0 lớn hơn 50% thì ước tính thẻ chưa đọc được tính bằng Công thức (15)

$$n' = n - S - (E_0)F \quad (15)$$

trong đó, n' là kết quả ước tính thẻ mới sau khi thực hiện hai phần trên và thiết lập n' là kích thước khung mới cho chu kỳ tiếp theo. E_0, S_0, C_0 là độ lệch giữa xác suất trung bình tối ưu và xác suất trong một khung hiện tại tương ứng với số khe xung đột, số khe rỗi và số khe thành công. Các giá trị được tính theo các Công thức (16), (17), (18):

$$|B_0 - OES| = E_0 \Rightarrow E_0 = |B_0 - 0.37| \quad (1)$$

$$|B_1 - OSS| = S_0 \Rightarrow S_0 = |B_1 - 0.37| \quad (2)$$

$$|B_k - OCS| = C_0 \Rightarrow C_0 = |B_k - 0.26| \quad (3)$$

Trong 2CTE, việc ước tính thẻ dựa trên xác suất trung bình tối ưu của số khe thành công, số khe rỗi và số khe xung đột. Kỹ thuật này ước tính thẻ và thay đổi kích thước khung tương đối gần bằng với số lượng thẻ chưa đọc trên thực tế; bên cạnh việc giảm xung đột còn tính đến việc giảm khe rỗi để tránh lãng phí khe thời gian, từ đó cho thấy tổng số khe dùng để nhận dạng thẻ là thấp hơn DFSA, EDFSA và cho hiệu quả hệ thống tốt.

Một so sánh giữa các kỹ thuật ước tính thẻ RFID trong giao thức DFSA được thể hiện trong Bảng 3.

Bảng 3. So sánh các ưu điểm và nhược điểm của các kỹ thuật ước tính thẻ.

Giao thức/ Kỹ thuật	Đặc điểm	Ưu điểm	Nhược điểm
DFSA	Thay đổi kích thước khung	Giảm số khe xung đột	Tăng số khe rỗi
EDFSA	Nhóm thẻ; hạn chế số thẻ trả lời	Giảm số khe xung đột	Tăng số khe thời gian
2CTE	Cân bằng số khe xung đột giảm	Giảm thời gian để	Tăng xung đột nếu số

Giao thức/ Kỹ thuật	Đặc điểm	Ưu điểm	Nhược điểm
	và số khe rỗi giảm	nhận dạng hết tất cả các thẻ	thẻ cần đọc lớn

4. SO SÁNH VÀ ĐÁNH GIÁ DỰA TRÊN MÔ PHỎNG

Các thuật toán được cài đặt bằng ngôn ngữ Java trên PC CPU Intel Core i3-2330M. 2.20GHZ, 4GB RAM. Xét các trường hợp số thẻ cần nhận dạng thay đổi từ 50 đến 100 thẻ. Để giảm sự phức tạp của quá trình mô phỏng, giả định rằng không có bất kỳ thẻ nào được thêm vào hoặc rời đi khỏi vùng đọc trong các chu kỳ đọc. Mục tiêu mô phỏng là so sánh các giá trị SE, TE, TS, TC, NC, NS của các kỹ thuật EDFSA, 2CTE với DFSA.

Các tiêu chí so sánh gồm:

– SE: Hiệu quả hệ thống được đánh giá dựa vào kết quả số khe rỗi, số khe thành công và số khe xung đột, được tính như trong [4]:

$$SE = \frac{TS}{TS + TE + TC}$$

– TE: Tổng số khe rỗi trong quá trình nhận dạng thẻ.

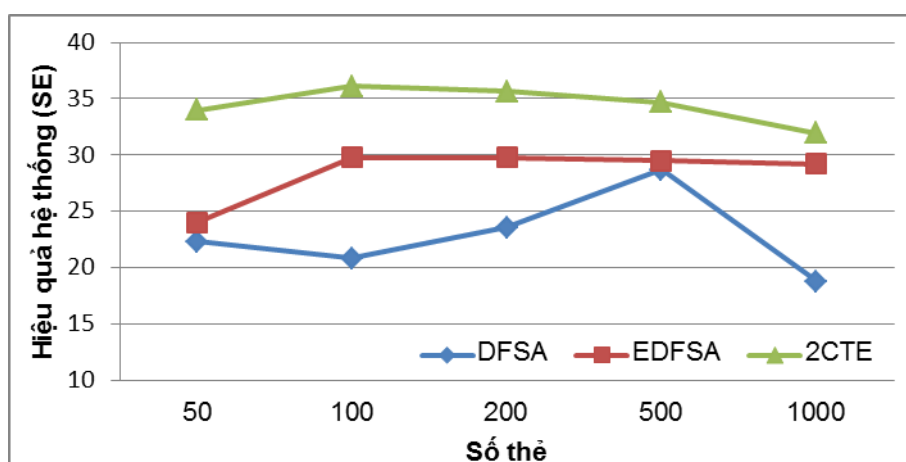
– TS: Tổng số thẻ đọc thành công trong quá trình nhận dạng thẻ, hay nói cách khác là tổng số khe thành công trong quá trình nhận dạng thẻ.

– TC: Tổng số khe xung đột trong quá trình nhận dạng thẻ.

– NC: Tổng số chu kỳ dùng để nhận dạng các thẻ trong vùng phủ sóng của đầu đọc

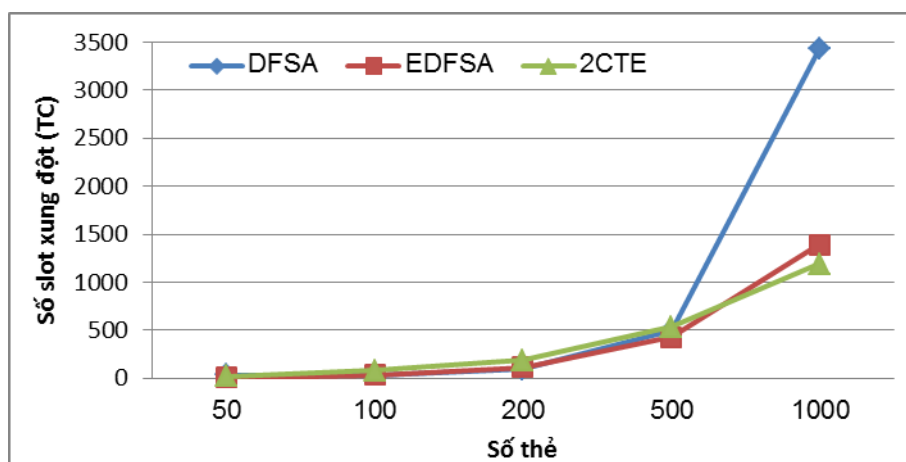
– NS: Tổng số khe thời gian mà đầu đọc dùng để nhận dạng được hết thẻ.

Xét về hiệu quả hệ thống (SE), kết quả mô phỏng trong Hình 3 cho thấy 2CTE đạt được hiệu quả cao nhất đối với tất cả các trường hợp thay đổi số lượng thẻ cần nhận dạng (từ 50 đến 1000 thẻ).



Hình 3. So sánh giá trị hiệu quả hệ thống (SE) với số thẻ cần nhận dạng thay đổi.

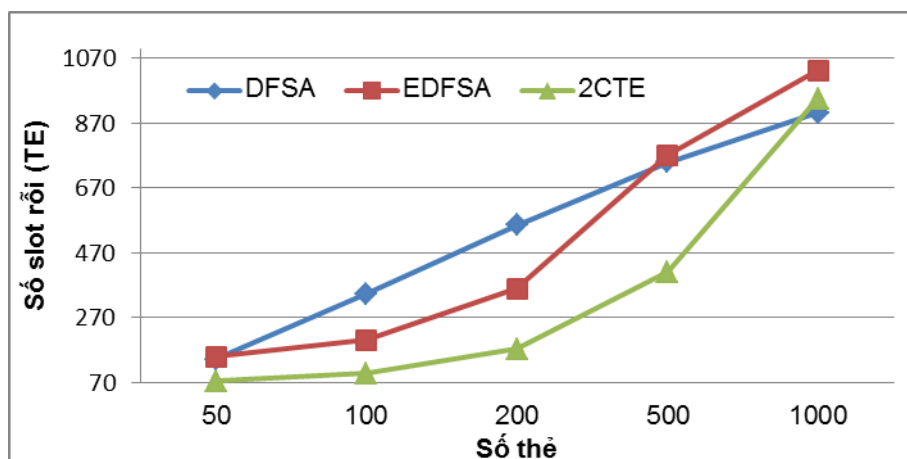
Khi xem xét đến các tiêu chí thành phần: TS, TE và TC, Hình 4 cho thấy khi số thẻ cần nhận dạng dưới 500, số khe xung đột của các kỹ thuật là tương đương nhau; nhưng khi số thẻ tăng cao (hơn 500), tổng số khe xung đột của 2CTE và EDFSA thấp hơn nhiều so với DSFA. Điều này cho thấy 2CTE và EDFSA giảm xung đột tốt hơn khi số thẻ cần nhận dạng tăng. Tuy nhiên, dựa vào công thức tính hiệu quả hệ thống (SE), để tăng hiệu quả hệ thống không chỉ phải giảm xung đột mà còn phải giảm số khe rỗi, bởi vì khi khe rỗi tăng sẽ làm tăng tổng thời gian nhận dạng tất cả các thẻ, gây lãng phí về mặt băng thông và thời gian.



Hình 4. So sánh tổng số slot xung đột (TC) với số thẻ cần nhận dạng thay đổi

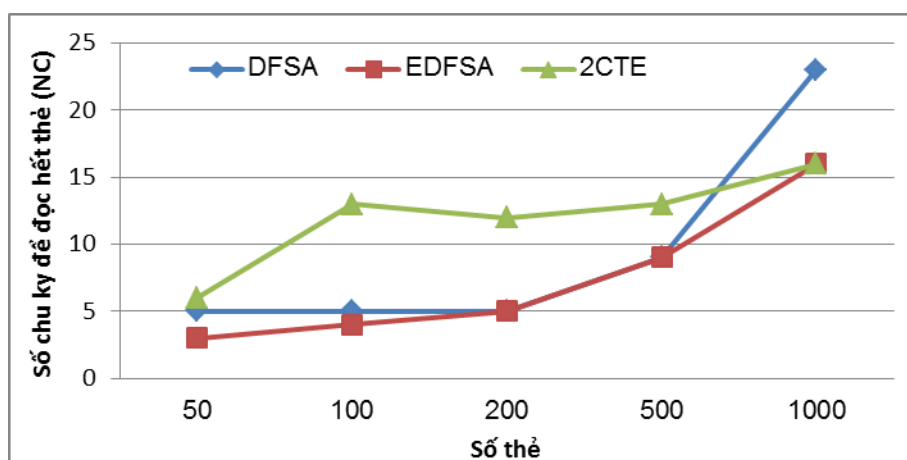
Trong Hình 5, EDFSA có tổng số khe rỗi tương đối thấp hơn so với DFSA khi số thẻ cần nhận dạng dưới 200 thẻ. Tuy nhiên, khi số thẻ cần nhận dạng trên 500 thẻ, số khe rỗi của EDFSA tăng cao do phải nhóm thẻ để hạn chế số thẻ trả lời đến đầu đọc. Điều này cho thấy EDFSA chỉ quan tâm đến việc giảm khe xung đột để tăng hiệu quả hệ thống, mà không quan tâm đến việc giảm khe rỗi. Trong khi đó, do 2CTE có quan tâm giảm khe rỗi nên có tổng số khe rỗi thấp hơn hẳn so với EDFSA và DFSA. Việc

giảm khe rỗi cũng được xem xét đồng thời với giảm khe xung đột nên khi số thẻ cần nhận dạng quá lớn (1000 thẻ), số khe rỗi của 2CTE sẽ cao hơn một ít so với DFSA, nhưng xét về hiệu quả hệ thống thì 2CTE vẫn tốt hơn do xung đột đã được giảm đáng kể.



Hình 5. So sánh tổng số khe rỗi (TE) với số thẻ cần nhận dạng thay đổi

Để giảm số khe xung đột và giảm số khe rỗi, 2CTE cần nhiều chu kỳ hơn so với DFSA và EDFSA để điều chỉnh kích thước khung cho phù hợp (Hình 6). Số chu kỳ là số lần đầu đọc gói khung đến các thẻ.

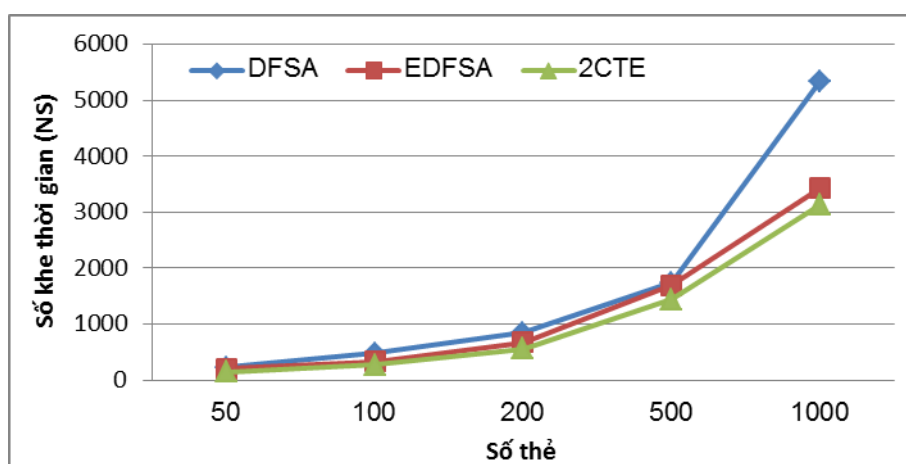


Hình 6. So sánh tổng số chu kỳ để đọc hết thẻ (NC) với số thẻ cần nhận dạng thay đổi

Với kỹ thuật DFSA và EDFSA, tổng số chu kỳ dùng để nhận dạng các thẻ thay đổi từ 50 thẻ đến 500 thẻ tương đối gần bằng nhau chỉ chênh lệch khoảng 1 đến 2 chu kỳ. Nhưng khi số thẻ cần nhận dạng lên đến 1000 thẻ, EDFSA sử dụng ít chu kỳ nhận dạng hơn do sử dụng nhóm thẻ nên số thẻ nhận dạng được trong mỗi chu kỳ là nhiều hơn DFSA.

Số chu kỳ nhận dạng của 2CTE tương đối nhiều hơn so với DFSA và EDFSA do cần phải ước tính thẻ nhiều lần để giảm xung đột và giảm số khe rỗi. Khi số thẻ nhận

dạng là 100 thẻ tổng số chu kỳ nhận dạng thẻ của 2CTE khoảng 13 chu kỳ trong khi DFSA và EDFSA chỉ cần khoảng 4 đến 5 chu kỳ, và khi số thẻ cần nhận dạng là 500 thẻ thì 2CTE cần khoảng 13 chu kỳ trong khi DFSA và EDFSA chỉ cần khoảng 9 chu kỳ. Tuy nhiên, điều này không ảnh hưởng nhiều đến hiệu quả hệ thống vì khi gởi khung thì kích thước khung cần gởi là bao nhiêu thì mới là vấn đề quan trọng. Kích thước khung là số khe thời gian dùng để nhận dạng thẻ, số khe thời gian càng lớn đồng nghĩa với việc cần nhiều thời gian nhận dạng hết thẻ và bằng thông tiêu tốn cho việc gửi các khung. Do đó cần phân tích và so sánh kết quả tổng số khe thời gian của các giao thức và kỹ thuật (Hình 7).



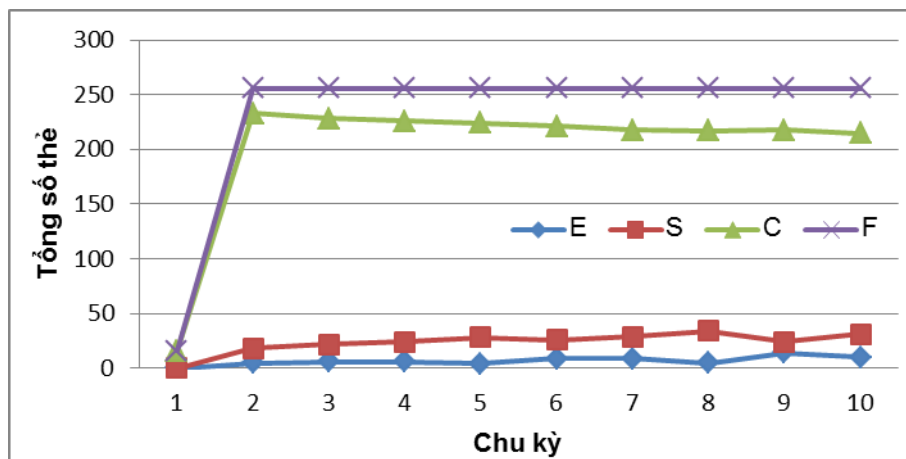
Hình 7. So sánh tổng số khe thời gian (NS) với số thẻ cần nhận dạng thay đổi

Khi số thẻ cần nhận dạng từ 50 thẻ đến 500 thẻ, tổng số khe thời gian của 2CTE nhỏ hơn DFSA và EDFSA, nhưng không nhỏ hơn nhiều. Khi số thẻ cần nhận dạng là 1000 thẻ ta nhận thấy DFSA cần rất nhiều khe thời gian để nhận dạng được hết thẻ so với EDFSA và 2CTE, tổng số khe thời gian của DFSA trong trường hợp này lên đến 5328 (Hình 7). Đối với EDFSA và 2CTE, tổng số khe thời gian của 2CTE nhỏ hơn EDFSA nhưng cũng không nhiều lắm chỉ chênh lệch khoảng 293 khe thời gian, đối với hiệu quả hệ thống của EDFSA và 2CTE cũng chênh lệch khoảng từ 2% - 3% (Hình 3). Qua đó cho thấy thời gian để nhận dạng được hết thẻ của 2CTE là ít hơn so với EDFSA và DFSA.

Để thấy rõ hơn hoạt động của các kỹ thuật ước tính thẻ, các tiêu chí như số khe rỗi (E), số khe thành công (S), số khe xung đột (C) và kích thước khung (F) trong mỗi chu kỳ của DFSA, EDFSA và 2CTE sẽ được tiếp tục được phân tích và so sánh. Mô phỏng được thực hiện với 1000 thẻ và các kết quả được trích xuất trong 10 chu kỳ đầu tiên.

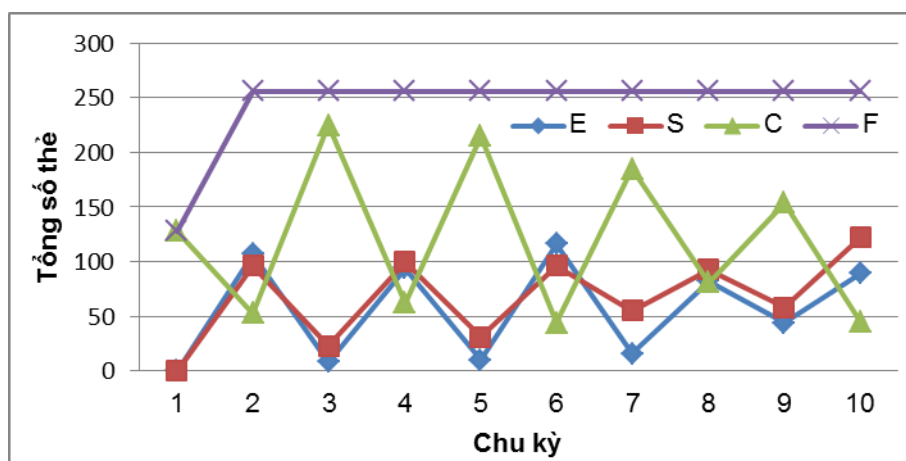
Với kỹ thuật DFSA, Hình 8 cho thấy sau chu kỳ thứ nhất, DFSA ước tính thẻ và thay đổi kích thước khung lên tối đa và liên tục trong 10 chu kỳ để cố gắng nhận dạng hết thẻ, do kích thước khung tối đa trong DFSA là 256 nhỏ hơn nhiều so với số thẻ cần

nhận dạng nên xung đột tăng cao gần bằng với kích thước khung, số khe rỗi và số khe thành công rất ít trong 10 chu kỳ đầu. Từ đó cho thấy hiệu quả hệ thống của DFSA trong trường hợp này là không cao.



Hình 8. So sánh giá trị E, S, C và F của giao thức DFSA

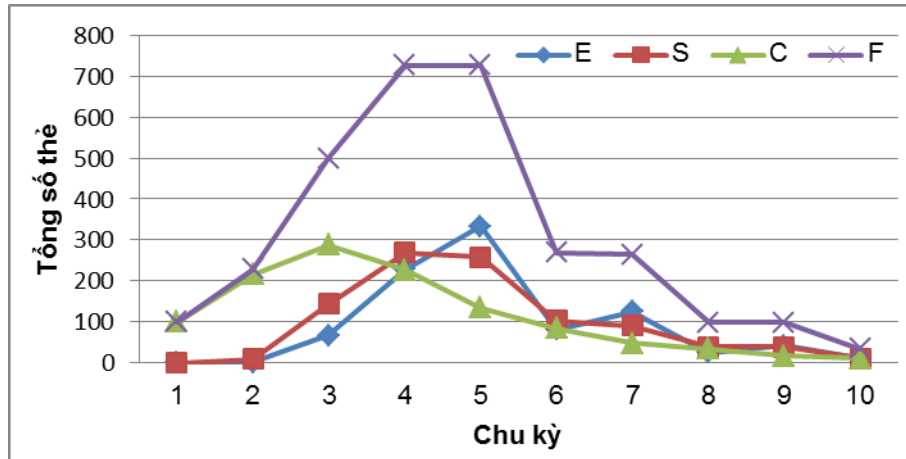
Để khắc phục nhược điểm của DFSA, EDFSA sử dụng phương pháp nhóm thẻ. Hình 9 cho thấy kích thước khung của EDFSA vẫn đặt lên tối đa từ chu kỳ 2 đến chu kỳ 10, nhưng tại các chu kỳ 2, 4, 6, 8, 10 xung đột giảm mạnh do EDFSA sử dụng phương pháp nhóm thẻ và số khe rỗi lại tăng cao do hạn chế số thẻ trả lời đến đầu đọc. Số thẻ đọc thành công tăng nhiều khi nhóm thẻ. Tại các chu kỳ còn lại không sử dụng phương pháp nhóm thẻ vì trong chu kỳ trước đã nhóm thẻ và đầu đọc ước tính thẻ dựa trên kết quả bộ ba $\langle E, S, C \rangle$, kết quả ước tính thẻ không lớn hơn 354, do đó xung đột vẫn tăng cao và số thẻ đọc thành công thì rất thấp.



Hình 9. So sánh giá trị E, S, C và F của kỹ thuật EDFSA

Với kỹ thuật 2CTE, Hình 10 cho thấy 2CTE ước tính thẻ tốt hơn và thiết lập kích thước khung ước tính cho chu kỳ tiếp theo. Trong các chu kỳ đầu 2CTE cố gắng giảm khe xung đột và số thẻ đọc thành công tăng từ chu kỳ 2 đến chu kỳ 5, số khe xung đột

đã giảm mạnh tại chu kỳ 5, nhưng số khe rỗi tăng cao tại chu kỳ này, do đó 2CTE thực hiện giảm kích thước khung và giảm khe rỗi từ chu kỳ 6. Từ chu kỳ 7 đến chu kỳ 10 ta nhận thấy kích thước khung, số khe rỗi và số khe xung đột bắt đầu giảm do 2CTE ước tính thẻ chính xác và vừa thực hiện giảm khe xung đột và khe rỗi. Từ đó cho thấy hiệu quả hệ thống của 2CTE cao hơn DFSA và EDFSA.



Hình 10. So sánh giá trị E, S, C và F của kỹ thuật 2CTE

5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Bài báo này đã phân tích và so sánh các kỹ thuật ước tính thẻ RFID trong giao thức DFSA, trong đó kỹ thuật 2CTE cho hiệu quả hệ thống tốt nhất. Tuy nhiên, các đề xuất này cũng bộc lộ các hạn chế như xung đột xảy ra vẫn cao khi số thẻ trong vùng đọc lớn, số khe rỗi vẫn nhiều và việc thay đổi kích thước khung liên tục ở mỗi chu kỳ đã làm tiêu tốn nhiều tài nguyên phần cứng. Do vậy, các hướng tiếp cận khác như vận dụng các phương pháp tính toán heuristics vào việc ước tính số thẻ chưa đọc trong vùng đọc để tính toán linh hoạt và chính xác hơn, nghiên cứu các cơ chế điều khiển thẻ đăng ký vào các khe rỗi của khung và nghiên cứu cơ chế phân cụm và phân lượt đăng ký linh hoạt đối với các thẻ.

TÀI LIỆU THAM KHẢO

- [1] Y. Huanjia and Y. Shuang-Hua , "RFID Sensor Network - Network Architectures to integrate RFID, sensor and WSN," *Measurement and Control*, vol. 40, no. 2, pp. 56-59, 2007.
- [2] L. Hai , B. Miodrag, N. Akshaya and S. Ivan, "Taxonomy and Challenges of the Integration of RFID and Wireless Sensor Networks," *IEEE Network*, vol. 22, no. 6, pp. 26-35, 2008.
- [3] L. Zhu and T.S.P. Yum, "A Critical Survey and Analysis of RFID Anti-Collision Mechanisms," *IEEE Communications Magazine*, vol. 49, issue 5, pp. 214-221, May 2011.

- [4] N. Abramson, "The aloha system: Another alternative for computer communications," in *Proceedings of Fall 1970 AFIPS fall joint computer conference*, pp. 281–285, 1970.
- [5] H. Okada, Y. Igarashi, and Y. Nakanishi, "Analysis and application of framed aloha channel in satellite packet switching networks-fadra method," *Electronics and Communications in Japan*, vol.60, pp. 72-80, 1977.
- [6] PHILIPS Semiconductor. I-CODE1 System Design Guide: Technical Report. May 2002.
- [7] K. Finkenzeller, RFID handbook - Second Edition, John Wiley & Sons, pp. 195–219, 2003.
- [8] H. Milad and A. Zahra, "2 Conditional tag estimation method for DFSA algorithms in RFID systems," in *International Conference on Computer and Knowledge Engineering (ICCCKE)*, Mashhad, Iran, 2014.
- [9] V. Harald, "Efficient Object Identification with Passive RFID Tags," *International Conference on Pervasive Computing*, pp. 98-13, 2002.
- [10] W. Jianwei, W. Dong and Z. Yuping, "A Novel Anti-Collision Algorithm with Dynamic Tag Number Estimation for RFID Systems," in *IEEE International Conference on Communication Technology*, Guilin, China, 2006.
- [11] L. Su-Ryun, J. Sung-Don and L. Chae-Woo, "An enhanced dynamic framed slotted ALOHA algorithm for RFID tag identification," in *The Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services*, San Diego, CA, USA, USA, 2005.
- [12] Š. Petar, R. Joško and R. Nikola, "Energy Efficient Tag Estimation Method for ALOHA-Based RFID Systems," *IEEE Sensors Journal*, vol. 14, no. 10, pp. 3637 - 3647, 2014.

ANALYSIS OF RFID TAG ESTIMATION TECHNIQUES IN DFSA PROTOCOL

Nguyen Van Vu*, Huynh Buu Ngoc

An Giang University

*Email: nvvu@agu.edu.vn

ABSTRACT

RFID is an automatic identification technology that is deployed in many practical applications today. In an RFID system, the reader identifies tags via radio waves without contact. A technical problem in RFID tag identification is that the conflict occurs when multiple tags are read simultaneously by a reader. Several estimation techniques have been proposed to reduce the conflict, minimize bandwidth

wastage, achieve low query time, and result in improved tag reading performance. This article will analyze some of the proposed tag estimation techniques in the DFSA protocol and compare them based on simulation results.

Keywords: DFSA, estimation techniques, RFID system, tag collision, system performance.



Nguyễn Văn Vũ sinh ngày 01/01/1986 tại An Giang. Năm 2008, ông tốt nghiệp đại học ngành Tin học tại trường Đại học An Giang; Năm 2016, tốt nghiệp thạc sĩ chuyên ngành Khoa học máy tính tại Trường Đại học Khoa học – Đại học Huế. Từ năm 2011 đến nay, công tác tại Phòng Đào tạo, Trường Đại học An Giang.



Huỳnh Bửu Ngọc sinh ngày 04/10/1983 tại An Giang. Năm 2007, ông tốt nghiệp kỹ sư Tin học tại Trường Đại học Cần Thơ. Năm 2010 đến nay, ông giảng dạy tại trường Trung học phổ thông Long Xuyên. Năm 2016 đến nay, ông theo học sau đại học chuyên ngành Khoa học máy tính tại trường Đại học Khoa học – Đại học Huế.