

ĐẠI HỌC QUỐC GIA HÀ NỘI

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

KHOA ĐIỆN TỬ VIỄN THÔNG

Báo cáo cuối kì

Đề tài: Xây dựng mô hình và đánh giá hiệu năng với phương thức
CSMA với module Wifi

Môn học: Mạng truyền thông máy tính 2

Sinh viên thực hiện:

Bùi Minh Hiếu (21021584)

Vũ Đình Nam(21021619)

Hoàng Đức Anh (21021555)

Giáo viên hướng dẫn:

TS. Lâm Sinh Công

Ngày 31 tháng 12 năm 2023



Mục lục

1	Lời mở đầu	2
2	Sơ lược	2
3	Phân tích về cơ chế CSMA	3
3.1	Giới thiệu về CSMA	3
3.2	Nguyên lý cơ bản của giao thức CSMA/CA:	3
3.3	Tại sao lại là CSMA/CA:	3
3.3.1	Các vấn đề gặp phải khi sử dụng giao thức CSMA tiêu chuẩn	4
3.3.2	Giải quyết vấn đề bằng cách sử dụng tín hiệu RTS và CTS	6
3.4	Nhận xét và kết luận:	7
4	Xây dựng mô phỏng	7
4.1	0: Nhàn rỗi	8
4.2	1: Đợi kênh truyền khả dụng	10
4.3	2: Kênh truyền đang khả dụng và access point đang đợi DIFS.	10
4.4	3: Đang gửi dữ liệu	12
4.5	4: Station đang trong thời gian đợi SIFS	12
4.6	5: Station gửi trả ACK	13
4.7	6: Chờ bản tin ACK	13
4.8	7: Nhận dữ liệu	14
4.9	-1: Đợi ngẫu nhiên do xung đột (random back-off)	15
5	Kết quả mô phỏng	16
A	Phụ lục	26

1 Lời mở đầu

Trong thời đại ngày nay, mạng không dây (Wi-Fi) đã trở thành một phần không thể thiếu của cuộc sống hàng ngày, cung cấp cho chúng ta sự kết nối và truy cập internet một cách linh hoạt và thuận tiện. Đằng sau sự hiệu quả và tiện lợi của mạng Wi-Fi là một loạt các giao thức và thuật toán phức tạp, trong đó CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) đóng vai trò quan trọng trong việc quản lý truy cập truyền thông không dây.

CSMA/CA là một phương pháp tránh va chạm được thiết kế để giảm thiểu xung đột dữ liệu trong môi trường truyền thông không dây. Trong bối cảnh mà hàng triệu thiết bị kết nối đồng thời, việc quản lý tài nguyên và đảm bảo hiệu suất cao là một thách thức đối với các hệ thống mạng Wi-Fi. Báo cáo này sẽ tập trung vào CSMA/CA và những khía cạnh quan trọng của nó trong việc tối ưu hóa hoạt động của mạng Wi-Fi.

Chúng ta sẽ khám phá cơ bản về CSMA nói chung và CSMA/CA nói riêng, cách nó hoạt động, và tại sao nó là một phần quan trọng của các tiêu chuẩn mạng không dây như IEEE 802.11. Bằng cách này, chúng ta có thể hiểu rõ hơn về cách giao thức giúp kiểm soát và phân phối băng thông trong môi trường mạng không dây đa dạng, đồng thời cung cấp các giải pháp cho các thách thức hiện đại mà hệ thống mạng Wi-Fi phải đối mặt.

2 Sơ lược

Người ta có thể coi giao thức CSMA là một sự phát triển của ALOHA trong đó các khả năng đầu cuối (sense) được khai thác để đạt được hiệu suất được cải thiện. Trên thực tế trong các hệ thống thực, khả năng đầu cuối cần thiết phụ thuộc vào phương tiện truyền dẫn, tức là "sense" được tận dụng phụ thuộc vào phương tiện truyền có thể là dây (cáp xoắn, đồng trục, quang) hay qua sóng vô tuyến trong khí quyển (giao tiếp không dây). Theo đó phần sau đây, chúng ta thảo luận về các giao thức CSMA và CSMA/CD thích hợp cho truyền thông có dây, mặt hạn chế của chúng và cách tiếp cận vấn đề đối với giao thức CSMA/CA được thiết kế cho truyền thông không dây.

3 Phân tích về cơ chế CSMA

3.1 Giới thiệu về CSMA

Carrier Sense Multiple Access (CSMA) là một nguyên tắc cơ bản trong quản lý truy cập kênh truyền thông trong mạng. Nó cho phép nhiều thiết bị chia sẻ cùng 1 kênh truyền (nên chúng có bản chất phân tán)

- **Dò tín hiệu (Carrier Sense):** Trước khi thiết bị bắt đầu truyền dữ liệu, nó sẽ kiểm tra xem kênh truyền đã được sử dụng hay chưa. Nếu kênh đang rảnh, thiết bị có thể bắt đầu truyền dữ liệu. Nếu kênh đang được sử dụng, thiết bị sẽ chờ cho đến khi kênh trở nên rảnh trước khi thử lại.
- **Truy cập ngẫu nhiên (Multiple Access):** Nếu nhiều thiết bị cùng thời điểm muốn truy cập kênh, CSMA cho phép chúng thử truy cập kênh ngẫu nhiên. Nguyên tắc này giúp giảm xác suất xung đột, nhưng không loại trừ khả năng xung đột xảy ra.
- **Xử lý xung đột (Collision):** Trong một số hình thức CSMA, nếu thiết bị phát hiện xung đột (một thiết bị bắt đầu truyền dữ liệu cùng một lúc với một thiết bị khác), nó sẽ bắt đầu thực hiện quy trình xử lý xung đột.

3.2 Nguyên lý cơ bản của giao thức CSMA/CA:

Trong điều khiển truy cập ngẫu nhiên thì CSMA/CA (Collision Avoidance) dùng để tránh xung đột dữ liệu từ nhiều thiết bị, CSMA/CA sử dụng chiến lược tránh xung đột thay vì phát hiện xung đột sau cùng như trong CSMA/CD (Collision Detection). Trước khi truyền, thiết bị kiểm tra lại kênh và truyền một khung nhỏ gọi là "Request to Send" (RTS). Nếu không có xung đột nào được phát hiện, thiết bị nhận được một phản hồi gọi là "Clear to Send" (CTS), và sau đó, bắt đầu truyền dữ liệu. Quá trình này giảm nguy cơ xung đột do nhiều thiết bị cố gắng truy cập kênh cùng một lúc.

3.3 Tại sao lại là CSMA/CA:

Bản chất phân tán của giao thức CSMA và độ trễ thấp mà nó gây ra khi số lượng thiết bị đầu cuối hoạt động nhỏ, nên đây là một phương thức tốt cho truyền thông. Tuy nhiên, những hạn chế nhất định trong môi trường không dây không cho phép thực hiện trực tiếp giao thức.

Về khả năng cảm nhận sóng mang của các thiết bị đầu cuối, mặc dù có thể nhưng không phải lúc nào có thể đảm bảo với xác suất cao rằng kênh đang trống (không có ai truyền). Để hiểu vấn đề này, chúng ta phải mở rộng các hạn chế đặc biệt được áp đặt trong môi trường không dây khi sử dụng loại CSMA tiêu chuẩn (basic) :

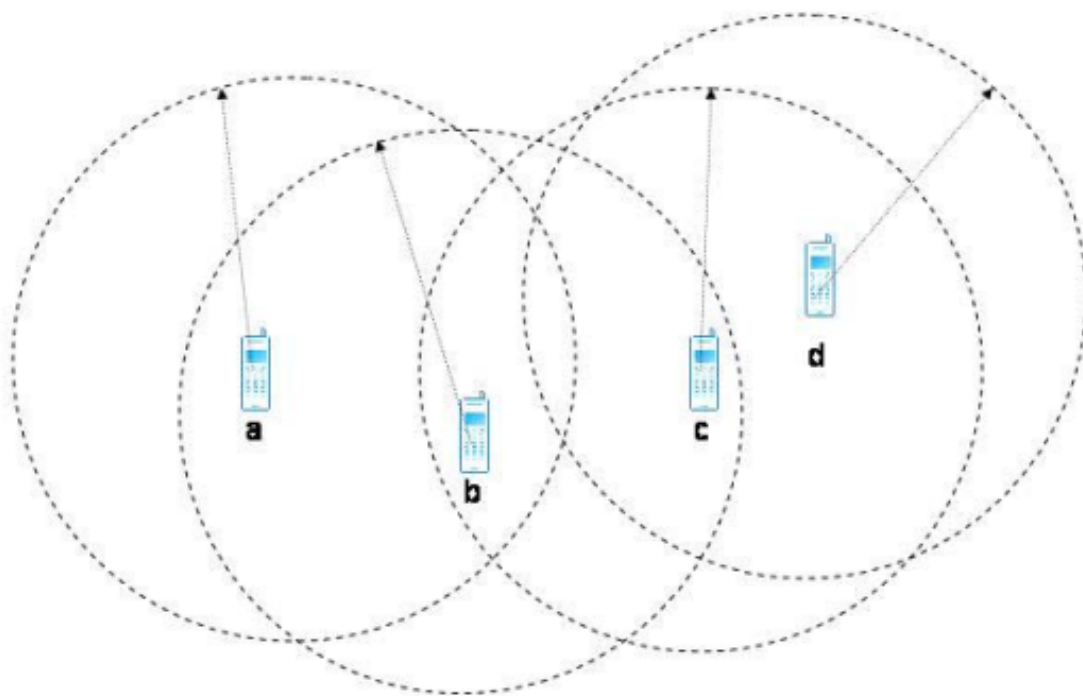
3.3.1 Các vấn đề gặp phải khi sử dụng giao thức CSMA tiêu chuẩn

Vấn đề 1 : Sự cố thiết bị đầu cuối ẩn

Một đặc điểm của truyền dẫn không dây là thiết bị đầu cuối (a) chỉ có thể cung cấp thông tin đáng tin cậy đến (b) nếu (b) nằm trong một khoảng cách nhất định so với (a).

Bây giờ hãy xem xét tình huống trong hình dưới đây trong đó chúng ta giả sử rằng việc truyền tải là đối xứng theo nghĩa là nếu thiết bị (a) có thể cung cấp thông tin đến (b) thì (b) có thể cung cấp thông tin đến a. Đường truyền từ (a) có thể đến được (b) nhưng không đến được (c). Đường truyền từ (c) có thể tới (b) nhưng không đến được (a). Sử dụng giao thức CSMA tiêu chuẩn trong môi trường này, vẫn có thể tránh được một số xung đột nhất định bằng cách cảm nhận kênh. Ví dụ: nếu (b) đang truyền tới (a), (c) có thể cảm nhận được quá trình truyền đang diễn ra. Tuy nhiên, giả sử rằng trong khi (a) truyền tới (b), (c) nhận được một gói để truyền. Nếu (c) lắng nghe kênh, nó sẽ không nghe thấy đường truyền của (a) và do đó, nếu giao thức CSMA tiêu chuẩn được sử dụng thì xung đột sẽ xảy ra.

Vấn đề này được gọi là vấn đề thiết bị đầu cuối ẩn. Lưu ý rằng trong trường hợp này, ngay cả khi (a) có thể phát hiện xung đột, nó sẽ không thể nhận ra rằng xung đột đã xảy ra vì nó không thể nghe thấy đường truyền (c) - như chúng ta đã thấy vấn đề sau được khắc phục bằng cách sử dụng thông báo ACK



Hình 1: Mô phỏng cho các thiết bị đầu cuối và khoảng cách truyền tín hiệu của chúng

Vấn đề 2 : Sự cố thiết bị đầu cuối bị lộ

Do chính sách truyền lại của giao thức CSMA cơ bản, hệ thống vẫn có thể hoạt động trong môi trường này mặc dù số lượng xung đột tăng lên, tuy nhiên, thông lượng hệ thống có thể giảm đáng kể nếu kích thước gói lớn. Trong thực tế, cảm nhận sóng mang đơn giản không phải lúc nào cũng được mong muốn trong môi trường này. Để làm rõ điểm này, hãy xem xét lại tình huống trong hình trên. Giả sử (b) đang gửi dữ liệu đến (a) và (c) muốn gửi dữ liệu đến thiết bị đầu cuối (d). Nếu (c) cảm nhận được kênh, nó sẽ thấy kênh đó đang bận và do đó sẽ trì hoãn việc truyền tải. Tuy nhiên, vì quá trình truyền của (c) không thể đến được (a) nên trên thực tế, (c) có thể chuyển gói của nó tới (d) mà không xung đột với quá trình truyền của (b). Kết quả là, cảm biến sóng mang đơn giản trong trường hợp này dẫn đến việc giảm mức sử dụng hệ thống. Sự cố này được gọi là sự cố thiết bị đầu cuối bị lộ.

Tiếp theo xét đến một cơ chế khác để giải quyết các vấn đề nêu trên. Hai tín hiệu điều khiển được đưa vào. Các tín hiệu điều khiển này là các tín hiệu ngắt (so với kích thước gói) được trao đổi giữa máy phát và máy thu trước khi bắt đầu truyền gói. Tín hiệu điều khiển đầu tiên được máy phát gửi đến máy thu và cho biết rằng máy phát đang “yêu cầu gửi” (RTS - Request to send) một gói. Bộ thu, sau khi nhận được RTS chính xác, sẽ trả lời rằng gói tin “rõ ràng để gửi” (CTS - Clear to send) . Cả hai tín hiệu RTS và CTS đều bao gồm một trường cho biết thời gian truyền gói và thông báo ACK đi kèm sẽ kéo dài bao lâu. Các thiết bị đầu cuối hiện hoạt động như sau :

- Nếu một thiết bị đầu cuối nghe tín hiệu CTS, nó sẽ đợi cho đến khi kết thúc quá trình truyền đang diễn ra; điều này được biết vì nó được bao gồm trong tín hiệu CTS. Sau đó thiết bị đợi một khoảng thời gian ngẫu nhiên và cố gắng bắt đầu quá trình truyền của chính nó.

3.3.2 Giải quyết vấn đề bằng cách sử dụng tín hiệu RTS và CTS

Giải quyết vấn đề 1:

Giả sử hiện tại việc truyền tín hiệu CTS và RTS là tức thời và chúng ta quay lại tình huống trong hình trên, trong đó (a) cần truyền một gói đến (b). Đầu cuối (a) gửi RTS đến (b) và (b) trả lời bằng tín hiệu CTS. Thiết bị đầu cuối (c) nhận được tín hiệu CTS và biết rằng quá trình truyền đã được bắt đầu nên nó trì hoãn quá trình truyền của chính nó. Do đó các vấn đề về thiết bị đầu cuối ẩn được giảm bớt. Trong thực tế, việc trao đổi các bản tin CTS và RTS hoạt động như một cơ chế cảm nhận sóng mang ảo.

Giải quyết vấn đề 2:

Giả sử rằng chúng ta thêm quy tắc sau : Nếu một thiết bị đầu cuối nghe tín hiệu RTS chứ không phải CTS thì nó sẽ tiếp tục truyền tín hiệu của chính nó, nếu có. Trong hình trên, giả sử rằng (b) gửi RTS đến (a) và trả lời bằng CTS. Thiết bị đầu cuối (c) nghe thấy RTS từ (b) nhưng không nghe thấy CTS từ (a) và do đó nó biết rằng việc truyền tải của chính nó sẽ không ảnh hưởng đến (b) đến quá trình truyền tải. Do đó nó có thể bắt đầu truyền tải riêng của mình bất cứ lúc nào. Do đó, vấn đề thiết bị đầu cuối tiếp xúc được tránh.

3.4 Nhận xét và kết luận:

Hệ thống CSMA tiêu chuẩn (basic) được sửa đổi có nguyên tắc hoạt động được mô tả ở trên, có tên CSMA/CA, trong đó CA là viết tắt của Collision Avoidance (Tránh va chạm), biểu thị rằng các va chạm được tìm cách tránh chứ không phải là chúng được tránh hoàn toàn. Do chính sách truyền lại của hệ thống CSMA, các xung đột có thể xảy ra không gây bất lợi: trong trường hợp xung đột, bản tin ACK hoặc bản tin RTS CTS sẽ không được nhận và thiết bị đầu cuối phát sẽ trì hoãn việc truyền nó trong một thời gian sau đó. Tuy nhiên, nếu độ trễ truyền tương đối lớn và hệ thống bị tải nặng thì xung đột có thể làm giảm hiệu suất của hệ thống.

4 Xây dựng mô phỏng

Mô phỏng mạng máy tính sử dụng giao thức đa truy cập CSMA trên MATLAB được đặt trong giả định vùng phủ sóng của tất cả các thiết bị đều bao quát tất cả các thiết bị còn lại, giúp chúng lắng nghe được sự kiện truyền của các thiết bị khác trên kênh truyền.

Mô phỏng được thực hiện trong báo cáo tập trung vào mô phỏng các sự kiện xảy ra trên kênh truyền của từng access point và station trong khoảng thời gian cố định.

Trong mô phỏng này, mỗi access point sẽ có xác suất cố định muốn gửi tới station một gói tin với kích thước cố định. Các access point và station tại 1 time slot có thể nhận 1 trong các trạng thái sau:

- 1: Đợi một khoảng thời gian ngẫu nhiên (đợi do xảy ra xung đột trên kênh truyền)
- 0: Nhận rồi (không có gói tin muốn truyền)
- 1: Đợi kênh truyền khả dụng (có gói tin muốn dùng nhưng kênh truyền đang bận)
- 2: Kênh truyền đang khả dụng và access point đang đợi DIFS.
- 3: Đang gửi dữ liệu
- 4: Station đang trong thời gian đợi SIFS trước khi gửi trả ACK.
- 5: Gửi bản tin ACK
- 6: Chờ bản tin ACK
- 7 : Nhận dữ liệu

Các hằng số được khởi tạo ban đầu của hệ được tính theo đơn vị time slot, với các parameter

được khởi tạo như sau:

- Kích cỡ frame: 10 time slots.
- Thời gian chờ SIFS: 2 time slots.
- Thời gian chờ DIFS: từ 3 đến 6 time slots.
- Thời gian mô phỏng: 500 time slots.
- Xác suất một access point có gói tin muốn truyền: 0.05.
- Thời gian back off khi gặp collision cơ sở: 4 time slots.
- Số lần back off tối đa: 8 lần.
- Mức độ ưu tiên của các trạm phát được khởi tạo ngẫu nhiên, là một trong các giá trị từ 1 đến 4 với cao nhất là 1 và thấp nhất là 4.

```

1 n=2;
2 simulation_time=1000;
3 frame_size=zeros(n+1,1)+10;
4 max_DIFS=6;
5 random_prob=0.05;
6 back_off_base=4;
7 max_back_off=8;

```

Trạng thái trong time slot tiếp theo của từng access point hoặc station phụ thuộc vào trạng thái của time slot hiện tại và kết quả sau khi lắng nghe kênh truyền.

4.1 0: Nhàn rỗi

Khí một access point (station) đang ở trạng thái *nhàn rỗi* ở thời điểm $t - 1$, ở time slot tiếp theo (thời điểm t) nó có thể chuyển sang các trạng thái sau:

- Nếu có một gói tin đang muốn truyền đến station và không có xung đột xảy ra, station sẽ chuyển sang trạng thái 7: *Nhận dữ liệu*.
- Nếu tại thời điểm t access point không có gói tin muốn truyền, nó sẽ tiếp tục duy trì trạng thái 0: *Nhàn rỗi*.

- Nếu tại thời điểm t access point có gói tin muốn truyền, nó sẽ lắng nghe xem kênh truyền có bận không, nếu kênh truyền đang bận, nó sẽ tiếp tục duy trì trạng thái 1: *Đợi kênh truyền khả dụng*. Trong trường hợp kênh truyền nhàn rỗi, trạng thái của access point sẽ chuyển thành trạng thái 2: *Kênh truyền khả dụng và access point đang đợi ACK*. Thời gian đợi DIFS sẽ là một số lượng time slot ngẫu nhiên trong khoảng xác định phụ thuộc vào độ ưu tiên của access point, cụ thể access point có mức độ ưu tiên cao hơn sẽ có tỉ lệ chờ DIFS thấp hơn so với các access point có mức độ ưu tiên thấp hơn.

```

1 case 0 %pre state is idle
2     for j=1:len          %check if we receive message
3         if(pre_status_matrix(j,2)==3&&first_frame_flag(j)==1)
4             frame_size(i)=frame_size(j);
5             count=count+1;
6             tmp=pre_comm_matrix(j);
7         end
8     end
9     if (count==1&&tmp==i)          % need to receive message
10        status(i)=7;          % go to ceceive state
11        timer(i)=frame_size(i);
12        comm(i)=pre_comm_matrix(i);
13    elseif(pre_comm_matrix(i)==0) % don need ack message check comm matrix
14        nothing to send stay idle
15        status(i)=0;
16        timer(i)=0;
17        comm(i)=pre_comm_matrix(i);
18    else %have something to send check if channel is busy
19        comm(i)=pre_comm_matrix(i);
20        for j=1:len
21            if(i~=j && pre_status_matrix(j,2)==3||pre_status_matrix(j,2)==5) %
22            channel is busy
23                status(i)=1;
24                timer(i)=0;
25                busy_flag=1;
26            end
27        end
28        if(busy_flag==0)          % channel is not busy

```

```

27         status(i)=2;
28         timer(i)=randi([max_DIFS-(3 - priority_matrix(i)) max_DIFS]);
29     end
30 end

```

4.2 1: Đợi kênh truyền khả dụng

Nếu tại thời điểm $t - 1$ access point đang đợi kênh truyền khả dụng thì tại thời điểm t :

- Nếu kênh truyền tiếp tục bận trong thời điểm t , access point sẽ tiếp tục đợi.
- Nếu kênh truyền khả dụng, access point sẽ chuyển sang trạng thái 2: *Kênh truyền khả dụng và đợi DIFS*

```

1 case 1 %pre state is waiting for media vailiable
2     for j=1:len
3         if(pre_status_matrix(j,2)==3||pre_status_matrix(j,2)==5) %channel is
        busy
4             status(i)=1;
5             timer(i)=0;
6         end
7     end
8     if(status(i)~=1) % channel is not busy
9         status(i)=2;
10        timer(i)=randi([max_DIFS-(3-priority_matrix(i)) max_DIFS]);
11    end

```

4.3 2: Kênh truyền đang khả dụng và access point đang đợi DIFS.

Nếu tại thời điểm $t - 1$ access point đang đợi DIFS thì tại thời điểm t :

- Nếu access point vẫn đang trong thời gian đợi DIFS và kênh không bận, nó sẽ tiếp tục đợi.
- Nếu kênh truyền bận, nó sẽ chuyển lại về trạng thái 1: *Đợi kênh truyền khả dụng*.
- Nếu access point đã kết thúc thời gian đợi DIFS và kênh không bận, nó sẽ bắt đầu gửi frame đến station.

```

1 case 2 %pre state is waiting DIFS
2     comm(i)=pre_comm_matrix(i);
3     if(pre_status_matrix(i,2)==3) % already start sending data
4         status(i)=3;
5         timer(i)=frame_size(i);
6     else
7         for j=1:len
8             if(i~=j && pre_status_matrix(j,2)==3 || pre_status_matrix(j,2)==5) %
channel is busy
9                 status(i)=1;
10                timer(i)=0;
11
12            end
13        end
14        if(status(i)~=1) % channel is not busy
15            status(i)=2;
16            timer(i)=pre_next_status_timer(i)-1;
17        end
18        if(status(i)==2 && timer(i)==0) %finish DIFS start sending Data
19            status(i)=3;
20            timer(i)=frame_size(i);
21            flag(i)=1;
22            plot_flag(i,1)=1;
23            plot_flag(i,2)=1;
24            plot_flag(i,3)=x_position(i);
25            plot_flag(i,4)=y_position(i);
26            plot_flag(i,5)=x_position(comm(i));
27            plot_flag(i,6)=y_position(comm(i));
28        end
29        for k=1:len %check if we receive message
30            if(pre_status_matrix(k,2)==3&&first_frame_flag(k)==1)
31                frame_size(i)=frame_size(k);
32                count=count+1;
33                tmp=pre_comm_matrix(k);
34            end
35        end
36        if (count==1&&tmp==i) % need to receive message

```

```

37         status(i)=7;                % go to ceceive state
38         timer(i)=frame_size(i);
39         comm(i)=pre_comm_matrix(i);
40     end
41 end

```

4.4 3: Đang gửi dữ liệu

Nếu tại thời điểm $t - 1$ access point đang gửi dữ liệu thì tại thời điểm t :

- Nếu frame chưa được truyền hết, access point sẽ tiếp tục gửi gói tin.
- Nếu frame đã được gửi hoàn tất, access point sẽ chuyển sang trạng thái 4: *Đợi SIFS (đợi để gửi ACK)*.

```

1 case 3 %pre state is sending data
2     comm(i)=pre_comm_matrix(i);    %keep sending and timer -1
3     status(i)=3;
4     timer(i)=pre_next_status_timer(i)-1;
5     flag(i)=0;
6
7     if(timer(i)==0)                %finished sending data, begin wait for ACK
8         status(i)=6;
9         timer(i)=4;
10    end

```

4.5 4: Station đang trong thời gian đợi SIFS

Nếu tại thời điểm $t - 1$ station đang đợi SIFS thì tại thời điểm t :

- Nếu thời gian đợi chưa kết thúc, access point sẽ đợi SIFS.
- Nếu kết thúc thời gian đợi, access point sẽ chuyển sang trạng thái 5: *Gửi bản tin ACK*.

```

1 case 4 %pre state is waiting for SIFS
2     comm(i)=pre_comm_matrix(i);
3     status(i)=4;
4     timer(i)=pre_next_status_timer(i)-1;

```

```

5     if(timer(i)==0)    %finished waiting SIFS
6         status(i)= 5;
7         timer(i)=2;
8     end

```

4.6 5: Station gửi trả ACK

Nếu tại thời điểm $t - 1$ station đang đợi kênh truyền khả dụng thì tại thời điểm t :

- Nếu gói tin ACK chưa được gửi hoàn tất, station sẽ tiếp tục gửi.
- Nếu gói tin ACK đã được gửi hoàn tất, station sẽ chuyển về trạng thái 0: *Nhàn rồi*.

```

1 case 5 %pre state is sending ACK
2     comm(i)=pre_comm_matrix(i);
3     status(i)=5;
4     timer(i)=pre_next_status_timer(i)-1;
5
6
7     if(timer(i)==0)    %finished sending ACK begin idle
8         status(i)= 0;
9         timer(i)=0;
10        comm(i)=pre_comm_matrix(i);
11    end

```

4.7 6: Chờ bản tin ACK

Nếu tại thời điểm $t - 1$ access point đang đợi bản tin ACK thì tại thời điểm t :

- Nếu bản tin ACK đã được nhận, access point sẽ chuyển về trạng thái 0: *Nhàn rồi*.
- Nếu vẫn còn trong thời gian đợi gói tin ACK, access point sẽ tiếp tục đợi.
- Nếu trong lúc gói tin ACK được truyền gặp xung đột với 1 gói tin khác, access point sẽ không xác nhận được nó đã gửi gói tin thành công. Lúc này access point sẽ chuyển sang trạng thái -1: *Đợi back off ngẫu nhiên*. Trong mô phỏng này, tình huống này sẽ không xảy ra do thời gian đợi DIFS để gói tin được xác nhận và gửi từ access point đến station dài hơn thời gian đợi SIFS.

```

1 case 6 %pre state is waiting for ACK
2     comm(i)=pre_comm_matrix(i);
3     status(i)=6;
4     timer(i)=pre_next_status_timer(i)-1;
5     if(timer(i)==0) %finished waiting for ACK
6         if(pre_status_matrix(comm(i),1)==5) % received ACK from the receiver end
begin idle
7             status(i)=0;
8             timer(i)=0;
9             comm(i)=0;
10            counter(i)=0; %flash counter when the data is successfully sended
11        else %collision happend random exponential back-off
12            status(i)=-1;
13            counter(i) = counter(i)+1;
14            timer(i) = randi([1 (back_off_base^counter(i))]); %random
exponential back-off
15            if(counter > max_back_off) %back-off exceed max value give up
sending data flash the counter
16                status(i)=0;
17                counter(i)=0;
18                timer(i)=0;
19                comm(i)=0;
20            end
21        end
22    end

```

4.8 7: Nhận dữ liệu

Nếu tại thời điểm $t - 1$ station đang nhận dữ liệu thì tại thời điểm t :

- Nếu gói tin vẫn chưa nhận đủ, station sẽ tiếp tục nhận dữ liệu
- Nếu gói tin đã được nhận đủ bởi station, station sẽ chuyển sang trạng thái 4: *Station đang trong thời gian SIFS trước khi trả ACK.*

```

1 case 7 %pre state is receiving message
2     comm(i)=pre_comm_matrix(i);

```

```

3     status(i)=7;
4     timer(i)=pre_next_status_timer(i)-1;
5     if(timer(i)==0)    %finished receiving begin SIFS and find out send ack back
to who
6         status(i)= 4;
7         timer(i)=2;
8         for j=1:len
9             if pre_status_matrix(j)==3
10                ack(i)=j;
11            end
12        end
13    end

```

4.9 -1: Đợi ngẫu nhiên do xung đột (random back-off)

Nếu tại thời điểm $t - 1$ access point đang trong quá trình back-off thì tại thời điểm t :

- Nếu thời gian đợi vẫn còn, access point sẽ tiếp tục đợi.
- Nếu thời gian đợi kết thúc, access point sẽ chuyển về trạng thái 0: *Nhàn rỗi*.

```

1 case -1 %pre state is Random back-off (Data collision)
2     comm(i)=pre_comm_matrix(i);
3     for j=1:len % check if channel is busy
4         if(pre_status_matrix(j,2)==3||pre_status_matrix(j,2)==5) %channel is
busy
5             status(i)=-1;
6             timer(i)=pre_next_status_timer(i);
7             busy_flag=1;
8         end
9     end
10
11     if(busy_flag==0)    % channel is not busy
12         status(i)=-1;
13         timer(i)=pre_next_status_timer(i)-1;
14     end
15
16     if(timer(i)==0) %finished random back-off

```



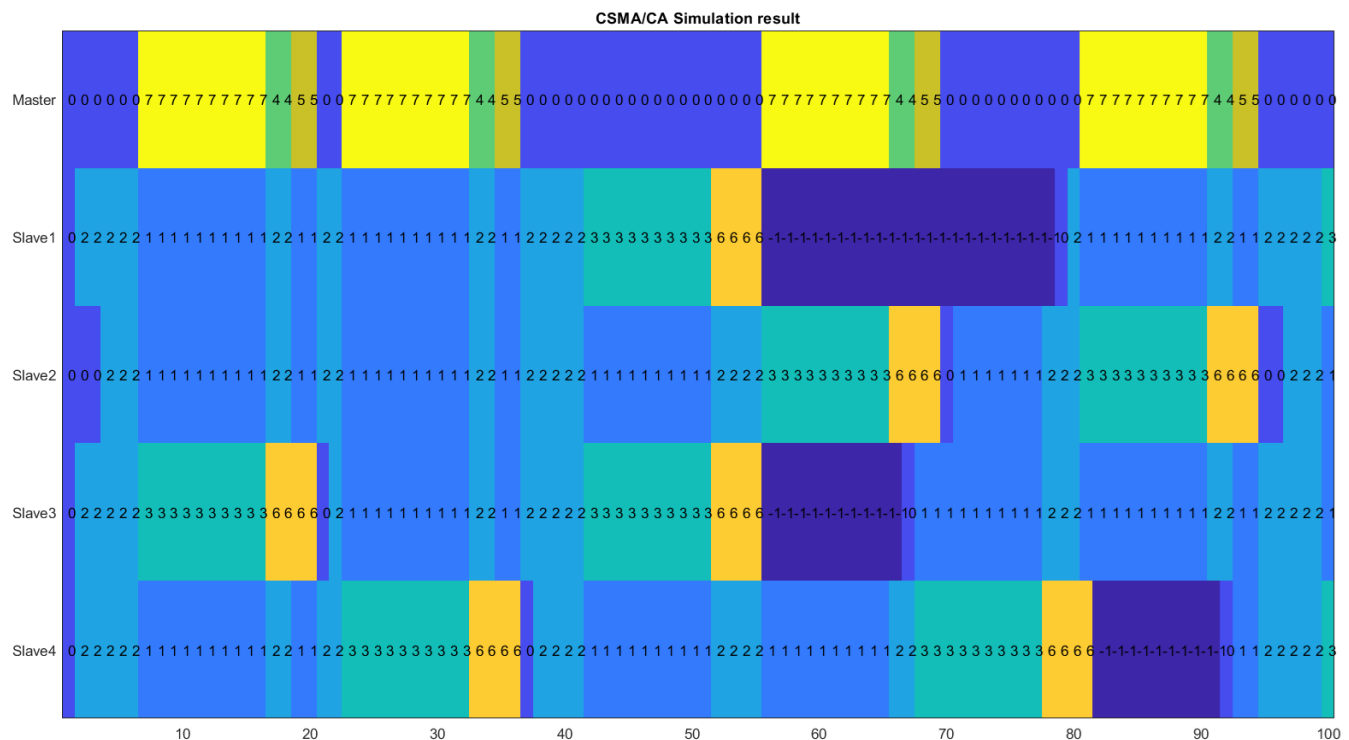
```

17         status(i)=0;
18         timer(i)=0;
19     end

```

5 Kết quả mô phỏng

Mô phỏng thực hiện với số access point thay đổi nhằm khảo sát một số đặc tính của mạng.



Hình 2: Ví dụ: Kết quả timeline một mô phỏng với 4 access point trong 100 time slot

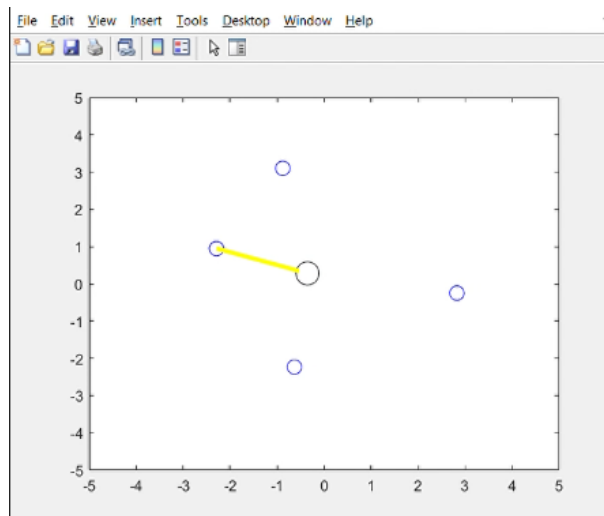
Thực hiện khảo sát các thông số tổng gói tin nhận được (thông lượng), tổng gói tin được gửi, số lần xảy ra va chạm, tỉ lệ thành công với số access point thay đổi từ 2 đến 10 trong khoảng thời gian 1000 timeslot, ta thu được các kết quả như sau.

Số access point	2	3	4	5	6	7	8	9	10
Tổng số gói tin được nhận	48	39	39	36	36	38	35	36	32
Tổng gói tin được gửi	52	66	71	78	82	79	92	95	102
Số lượng va chạm	2	9	15	19	19	17	19	20	26
Tỉ lệ thành công	92.31%	59.09%	54.93%	46.15%	43.90%	48.10%	38.04%	37.89%	31.37%

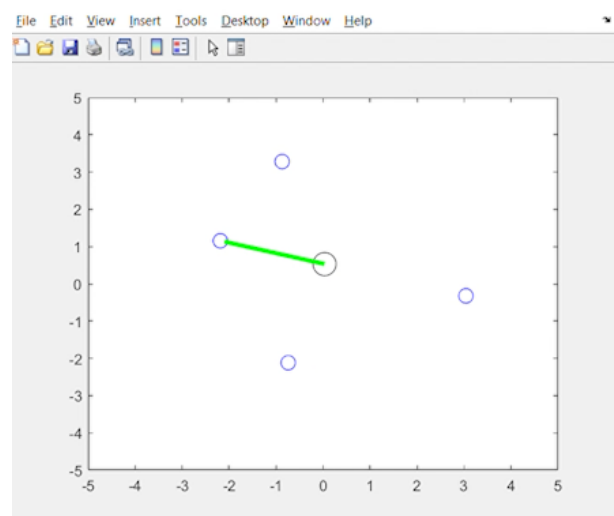
Bảng 1: Bảng thông số thu được sau mô phỏng

0	Nhàn rồi
1	Đợi kênh truyền khả dụng
2	Đợi DIFS
3	Gửi dữ liệu
4	Station đợi SIFS
5	Station gửi bản tin ACK
6	Access point đợi bản tin ACK
7	Nhận dữ liệu
-1	Thời gian Back-off

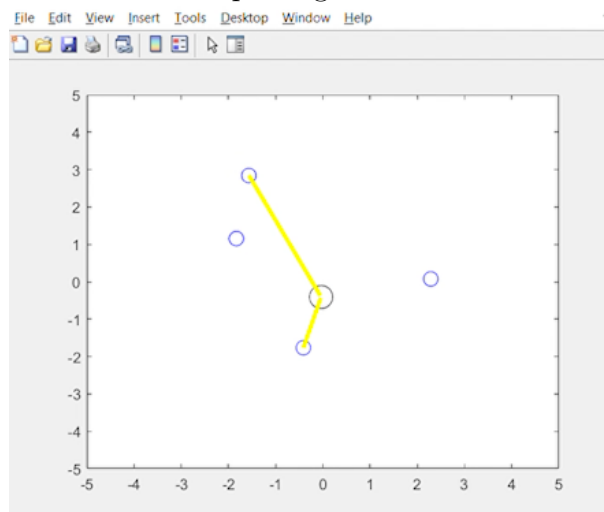
Hình 3: Chú thích ký hiệu



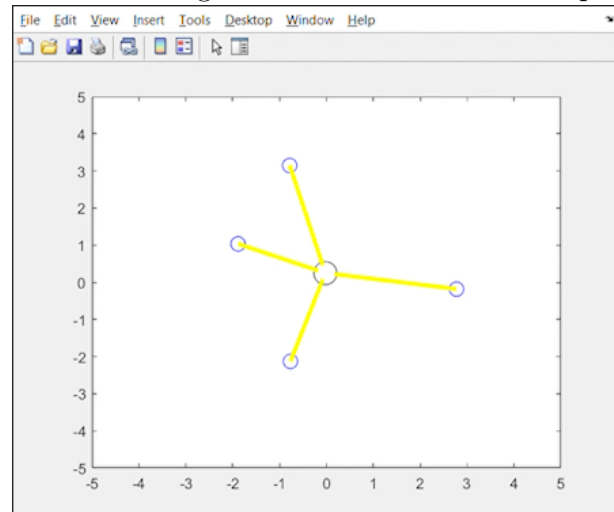
Hình 4: Access point gửi frame cho station



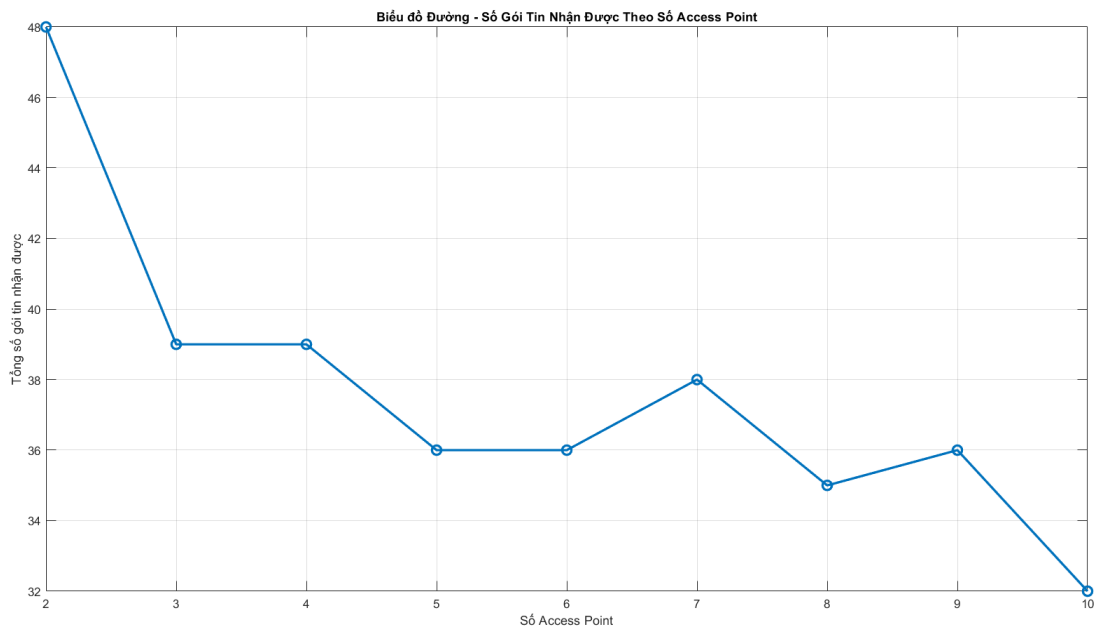
Hình 5: Station gửi bản tin ACK cho access point



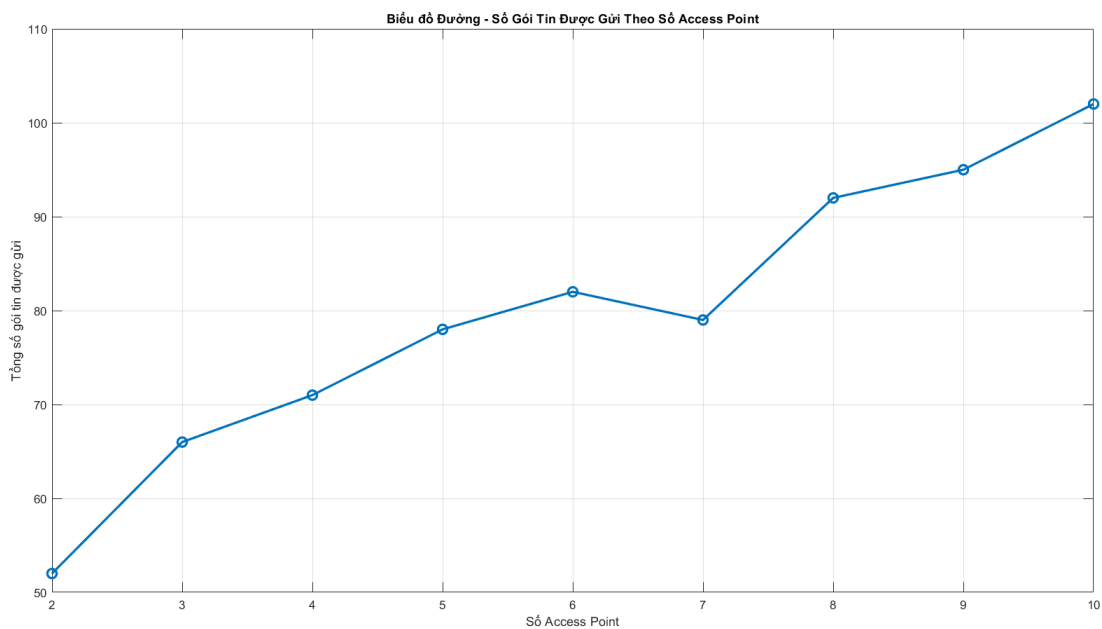
Hình 6: Xảy ra va chạm giữa 2 access point



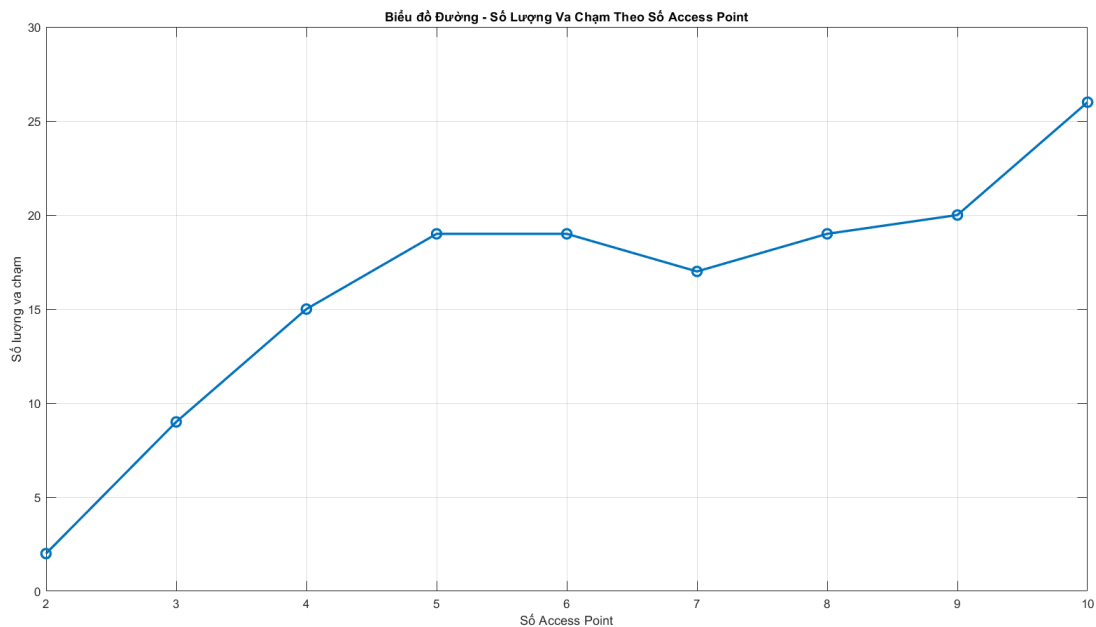
Hình 7: Xảy ra va chạm giữa nhiều access point



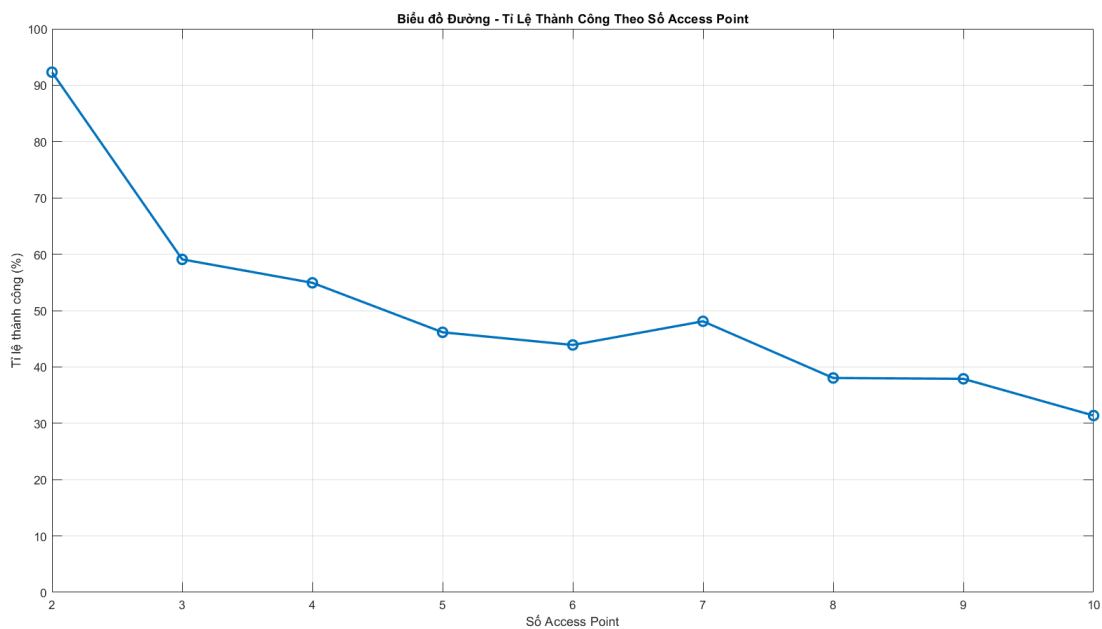
Hình 8: Số gói tin nhận được



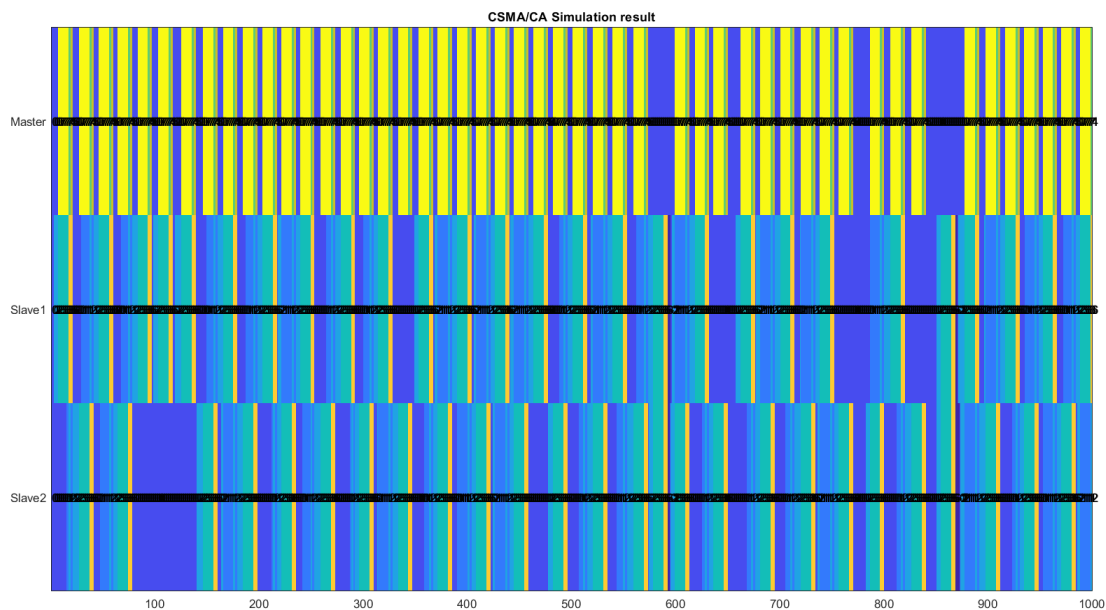
Hình 9: Số gói tin được gửi



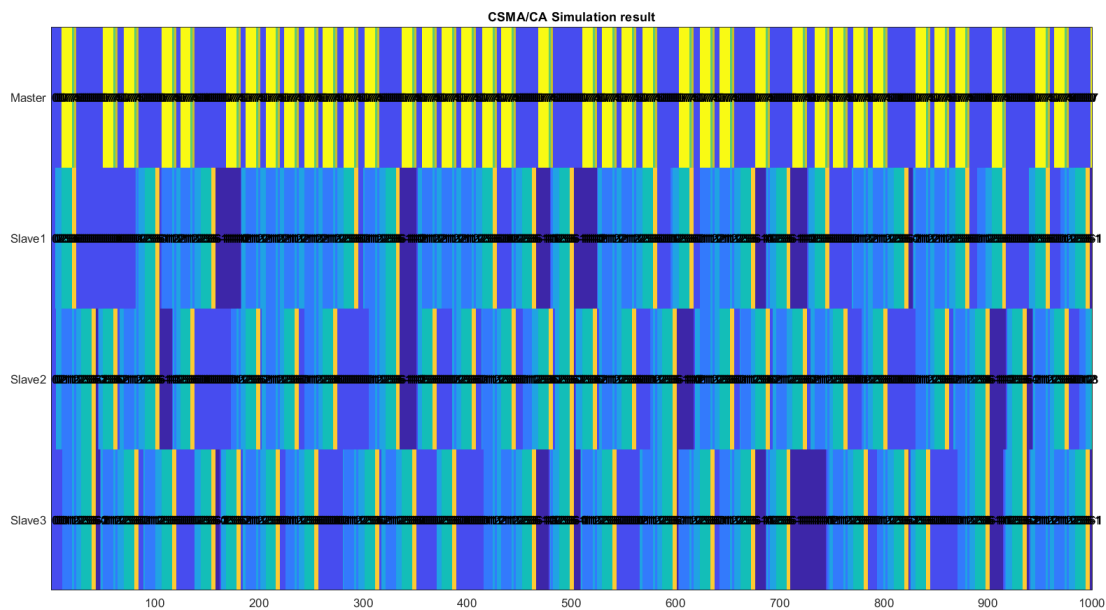
Hình 10: Số lần xảy ra va chạm



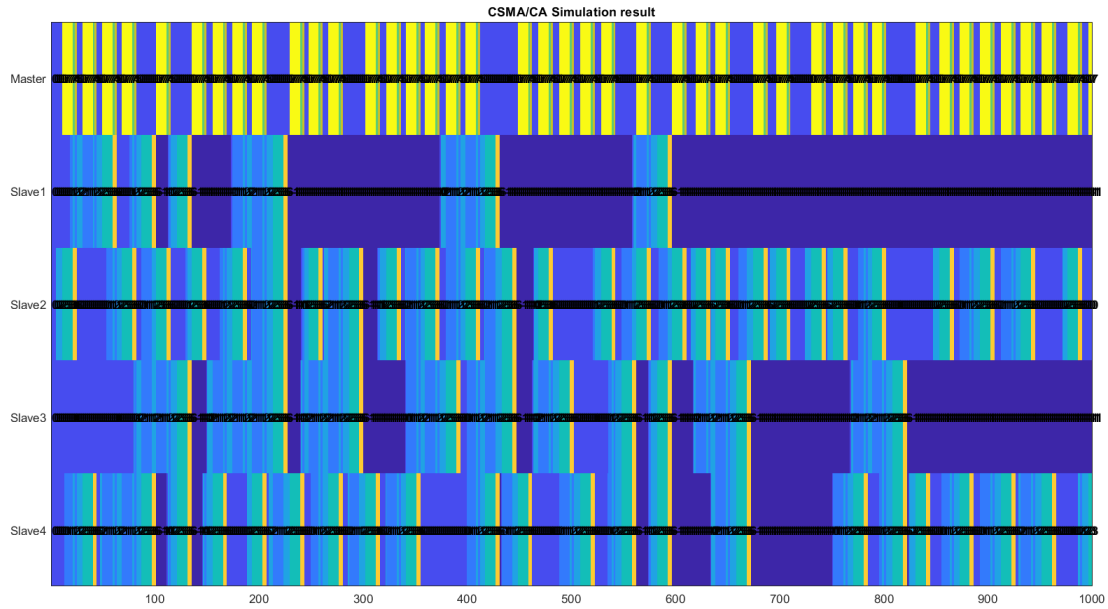
Hình 11: Tỷ lệ gửi thành công



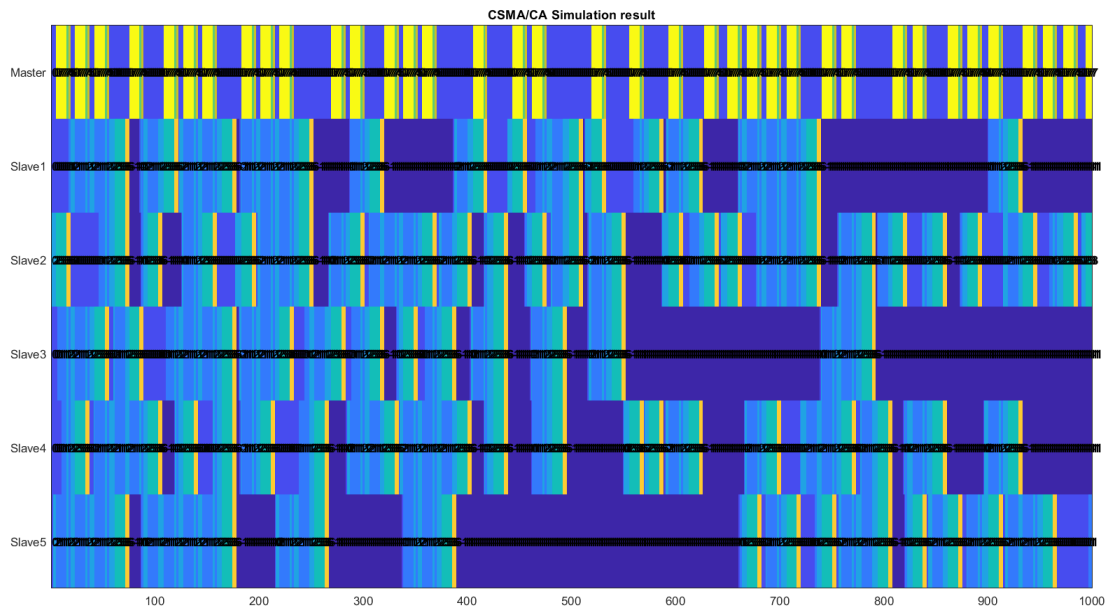
Hình 12: Kết quả mô phỏng với 2 access point



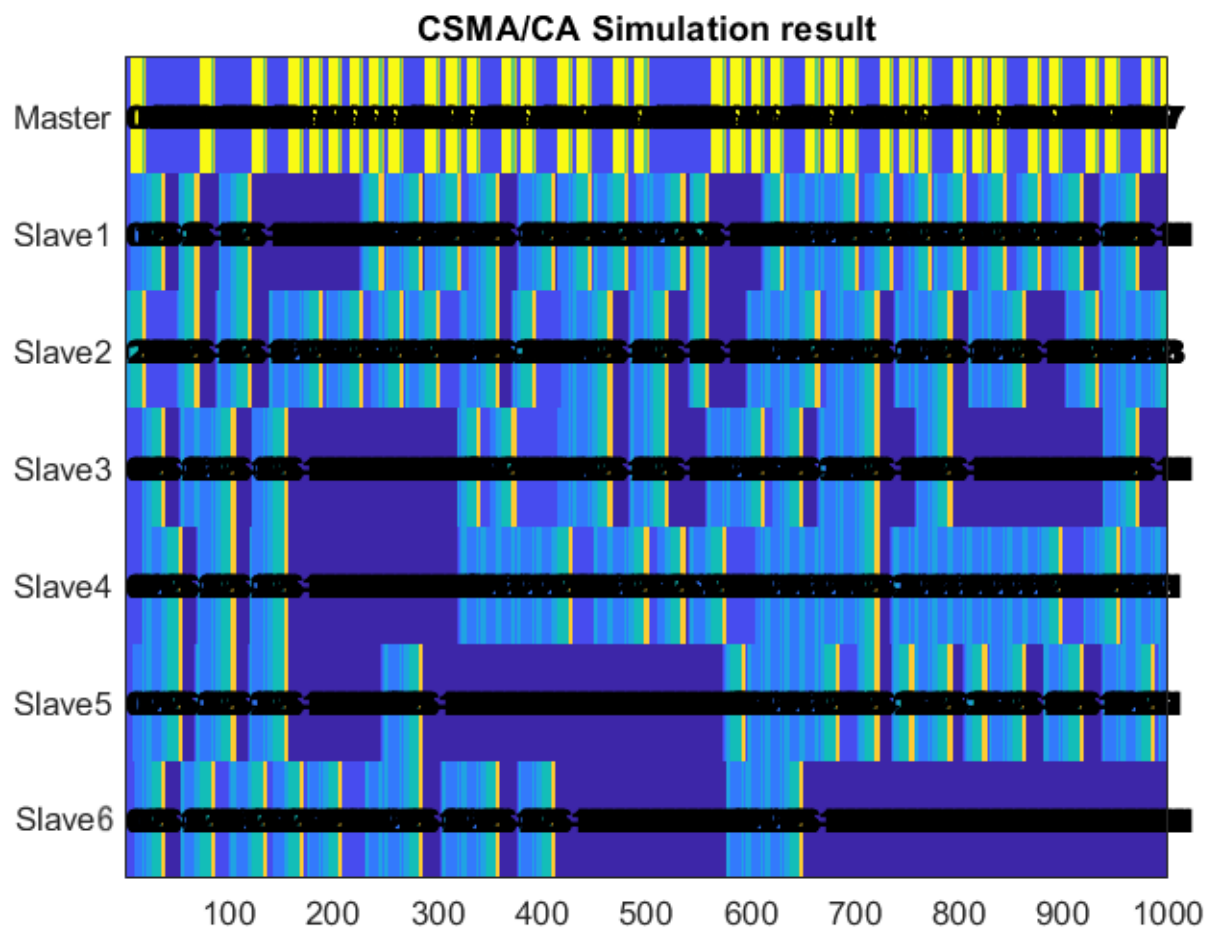
Hình 13: Kết quả mô phỏng với 3 access point



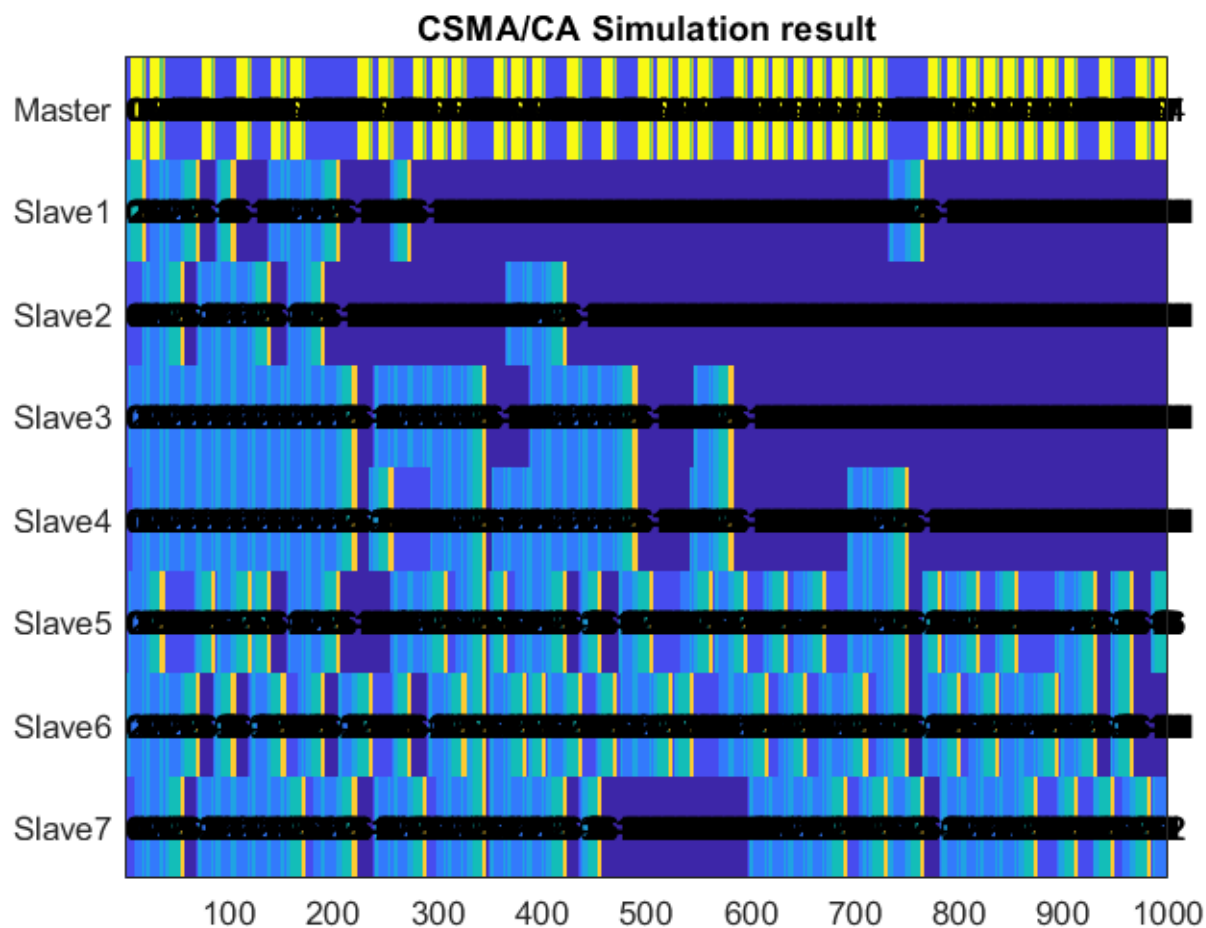
Hình 14: Kết quả mô phỏng với 4 access point



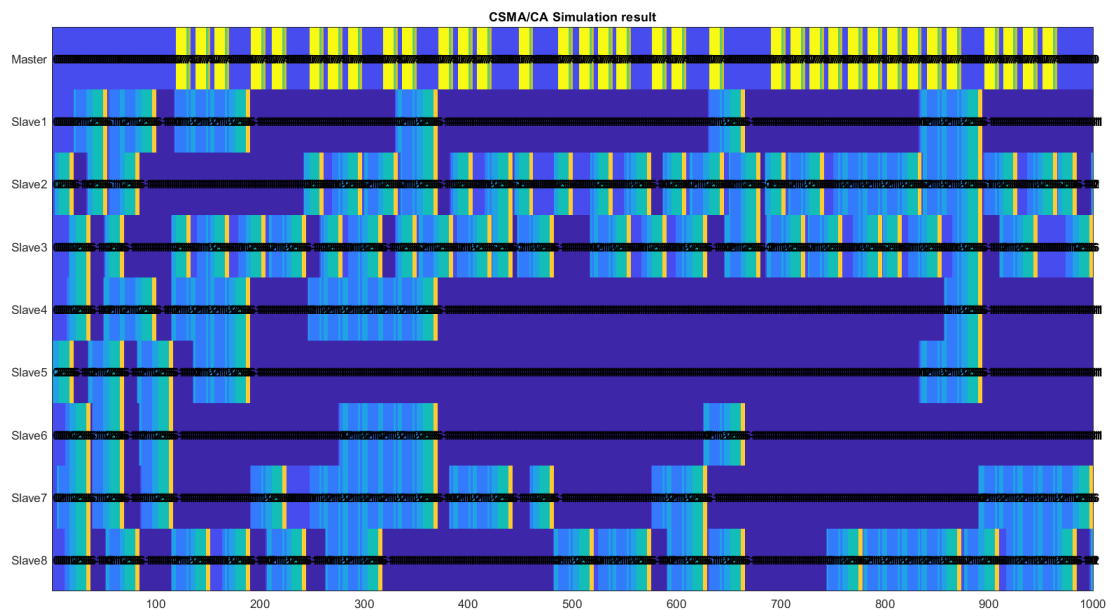
Hình 15: Kết quả mô phỏng với 5 access point



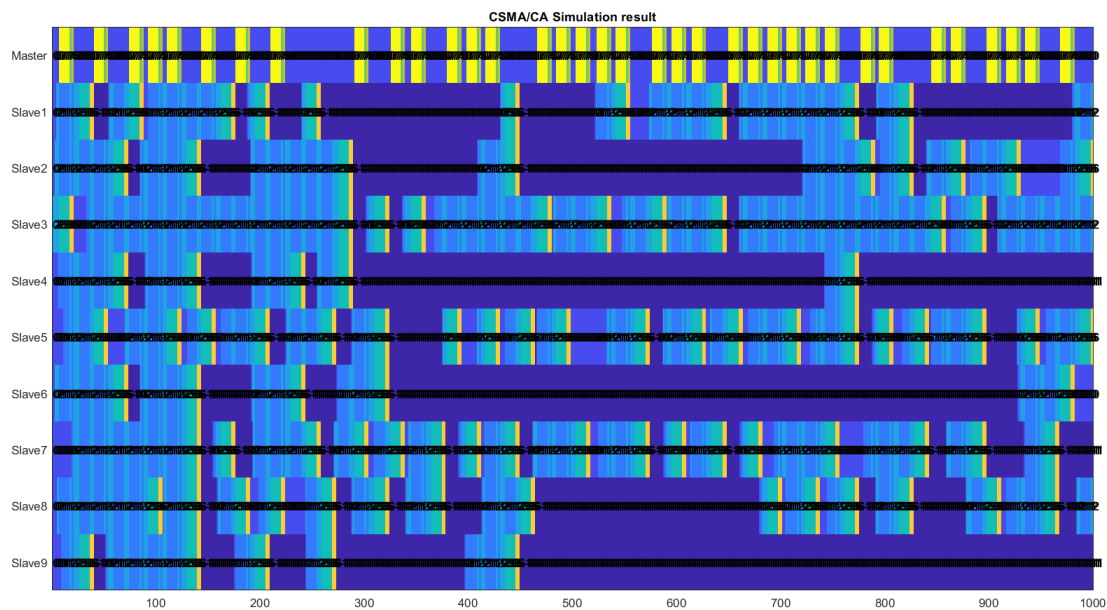
Hình 16: Kết quả mô phỏng với 6 access point



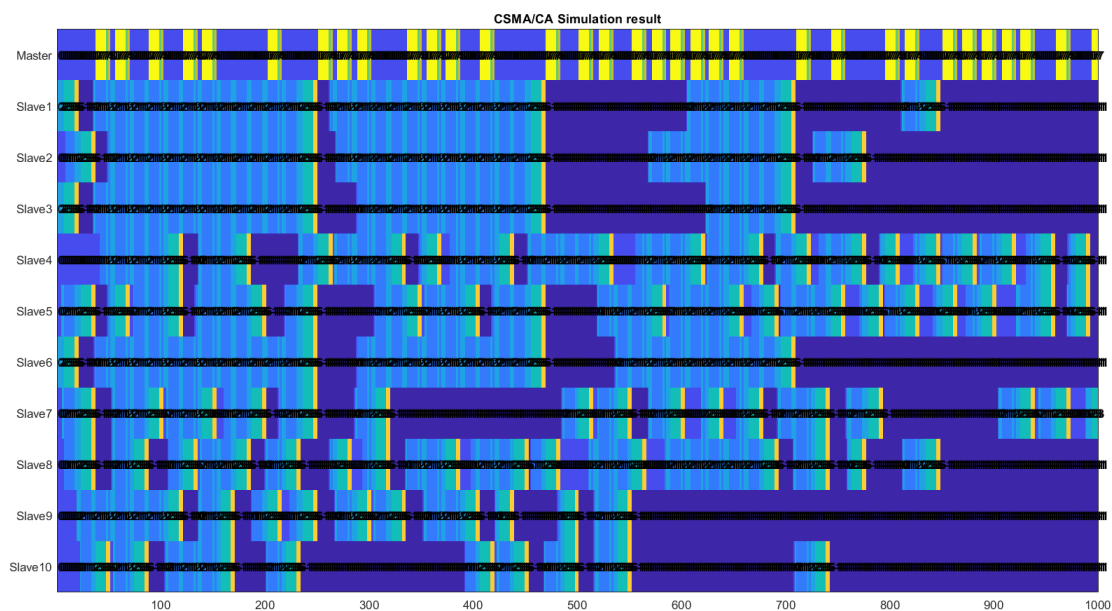
Hình 17: Kết quả mô phỏng với 7 access point



Hình 18: Kết quả mô phỏng với 8 access point



Hình 19: Kết quả mô phỏng với 9 access point



Hình 20: Kết quả mô phỏng với 10 access point

A Phụ lục

- [Link github của mô phỏng được sử dụng](#)