



(12) **BẢN MÔ TẢ GIẢI PHÁP HỮU ÍCH THUỘC BẰNG ĐỘC QUYỀN
GIẢI PHÁP HỮU ÍCH**

(19) **Cộng hòa xã hội chủ nghĩa Việt Nam (VN)
CỤC SỞ HỮU TRÍ TUỆ**

(11)



2-0004062

(51) **H04W 68/00**
2022.01

(13) **Y**

(21) 2-2023-00605

(22) 12/10/2023

(45) 26/05/2025 446

(43) 26/02/2024 431

(73) Đại học Bách khoa Hà Nội (VN)

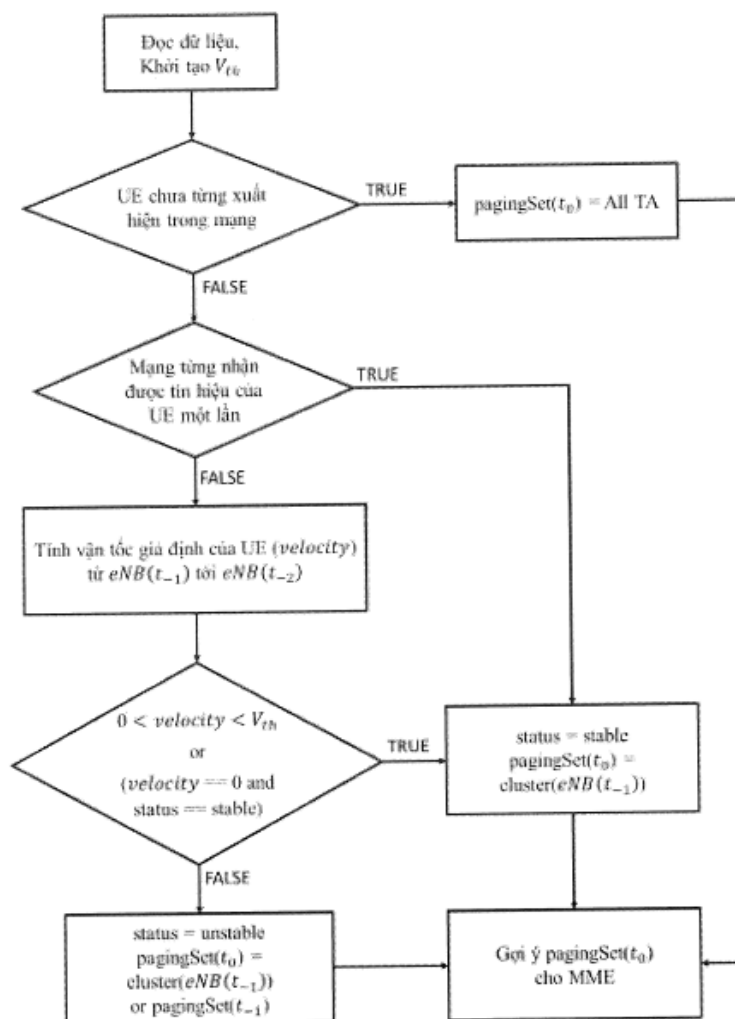
Số 1, Đại Cồ Việt, Hai Bà Trưng, Hà Nội

(72) PHÙNG THỊ KIỀU HÀ (VN); Nguyễn Tài Hưng (VN); Nguyễn Hữu Thanh (VN).

(54) **PHƯƠNG PHÁP GỌI NHẮN (PAGING) THIẾT BỊ NGƯỜI DÙNG (UE) TRONG
MẠNG TRUYỀN THÔNG DI ĐỘNG 4G DỰA TRÊN DỰ ĐOÁN VẬN TỐC UE**

(21) 2-2023-00605

(57) Giải pháp hữu ích đề xuất phương pháp gọi nhắc (paging) thiết bị người dùng (UE) trong mạng truyền thông di động 4G dựa trên dự đoán vận tốc UE. Đầu tiên UE mặc định sẽ trong trạng thái stable, khi nhận được thông tin của UE từ hệ thống, có thể tính được vận tốc giả định (velocity) của UE khi UE di chuyển từ eNB(t-2) tới eNB(t-1). Lúc này có hai trường hợp xảy ra: (i) vận tốc giả định lớn hơn 0 và nhỏ hơn vận tốc ngưỡng (V_{th}): UE vẫn nằm trong trạng thái ổn định (stable); (ii) vận tốc giả định lớn hơn vận tốc ngưỡng (V_{th}): UE rơi vào trạng thái không ổn định (unstable). Nếu vận tốc giả định của UE = 0 (UE không thay đổi eNB) thì UE vẫn trong trạng thái ngay trước đó của nó. UE chỉ chuyển từ không ổn định (unstable) sang ổn định (stable) khi vận tốc giả định lớn hơn 0 và nhỏ hơn vận tốc ngưỡng, chuyển từ ổn định (stable) sang không ổn định (unstable) khi vận tốc giả định lớn hơn vận tốc ngưỡng. Giải pháp sẽ gợi ý tập các eNB nằm trong bán kính R của eNB ngay trước đó cho MME khi UE đang ổn định (stable) và sẽ gợi ý thêm cả các eNB đã gợi ý trước đó cho MME nếu UE rơi vào không ổn định (unstable).



Hình 7

Lĩnh vực kỹ thuật được đề cập

Giải pháp hữu ích đề xuất phương pháp gọi nhắn (paging) thiết bị người dùng (UE) trong mạng truyền thông di động 4G dựa trên dự đoán vận tốc UE

Tình trạng kỹ thuật của giải pháp hữu ích

Trước hết, phần này sẽ giới thiệu khái quát về một mạng di động phổ biến hiện nay là mạng 4G cùng với đó là kiến trúc của mạng. Sau đó, sẽ trình bày khái niệm, vai trò và quá trình gọi nhắn (paging) UE trong mạng di động 4G.

Mạng di động 4G, còn được gọi là LTE (Long-Term Evolution) được đề xuất lần đầu tiên vào năm 2004 bởi Hiệp hội Viễn thông Di động Quốc tế (3GPP - 3rd Generation Partnership Project). 3GPP là một tổ chức liên minh giữa các tổ chức tiêu chuẩn viễn thông trên toàn cầu và đảm nhận vai trò quy định và phát triển các tiêu chuẩn di động cho các công nghệ di động, bao gồm 2G, 3G, 4G và 5G. Tuy mạng 5G đang được triển khai và phát triển mạnh mẽ nhưng mạng 4G vẫn là công nghệ di động chủ đạo và vẫn phục vụ hàng tỷ người dùng trên toàn cầu.

Kiến trúc mạng di động 4G

Kiến trúc đầu tiên của mạng di động 4G được đề xuất năm 2008 bởi 3GPP và được cập nhật liên tục kể từ thời điểm đó đến nay. Trong đó, mạng được chia làm hai thành phần chính là EPC (Evolved Packet Core – Lõi gói cải tiến) và E-UTRAN (Evolved Universal Terrestrial Radio Access Network - Mạng truy cập vô tuyến mặt đất đa năng tiên hóa). EPC là một phần của mạng lõi (CN - Core Network) - nơi chứa các công nghệ viễn thông khác ngoài LTE. EPC và E-UTRAN được liên kết với nhau thông qua một giao diện được gọi là S1.

Hình 1 thể hiện giao tiếp của UE với EPC thông qua một trong hai giao diện là NAS và AS. Một bản tin NAS có thể chứa thông tin về việc 'Update', 'Attach' hoặc 'Authentication', còn bản tin AS có thể chứa một gói dữ liệu. Có hai giao diện liên kết giữa EPC và UE (User Equipment - Thiết bị người dùng) là:

- NAS (Non Access Stratum – Tầng không truy cập): Kết nối trực tiếp từ UE tới EPC mà không cần tương tác với E-UTRAN.
- AS (Access Stratum – Tầng truy cập): Kết nối thông qua E-UTRAN do đó có thể được truy cập và sửa đổi bởi E-UTRAN.

Một số thành phần chính trong mạng 4G

E-UTRAN

E-UTRAN bao gồm một tập hợp các eNB (E-UTRAN Node B - trạm cơ sở trên mặt đất kết nối với UE). Các eNB được kết nối với nhau thông qua một giao diện được gọi là X2 và với EPC qua giao diện S1. Tuy nhiên, mỗi eNB chỉ có một số lượng kết nối X2 nhất định do vậy không phải mọi tương tác giữa hai eNB đều thông qua giao diện X2 giữa chúng mà có thể thông qua hai giao diện S1 với EPC. Khi eNB A muốn hủy kết nối X2 tới eNB B thì eNB B sẽ bắt buộc phải tuân theo.

Hình 2 thể hiện cho ta thấy các kết nối trong E-UTRAN. Ở đó, các eNB được kết nối với nhau thông qua giao diện X2 và với EPC - nơi chứa MME (Mobility Management Entity – Thực thể quản lý di động) thông qua giao diện S1.

Dưới đây là một số chức năng chính của các eNB trong E-UTRAN:

- Quản lý các tài nguyên vô tuyến.
- Lựa chọn MME thích hợp kết nối với UE khi không có thông tin về MME ưu tiên được cung cấp bởi UE.
- Định tuyến dữ liệu người dùng tới S-GW (Serving Gateway).
- Lập lịch và truyền tải tin nhắn paging, thông tin phát sóng và tin nhắn cảnh báo.
- Bàn giao các UEs khi đang trong trạng thái kết nối (connected) giữa các eNB khi có sẵn giao diện X2.

Mạng vô tuyến được chia thành các nhóm khác nhau ở nhiều cấp độ khác nhau. Một hoặc nhiều eNB sẽ nằm trong một TA (Tracking Area) và một eNB có thể chứa một hoặc nhiều Cells.

Như được thể hiện trên Hình 3, mỗi màu đại diện cho một TA và mỗi hình lục giác đại diện cho một Cell. Mỗi UE được đăng ký vào một danh sách các TA được gọi là TAI list (Tracking Area Identifier list – danh sách nhận dạng khu vực theo dõi) và UE có thể di chuyển giữa các TA trong list mà không cần thông báo với mạng rằng nó vừa di chuyển. Tuy nhiên, khi UE di chuyển ra khỏi tầm phủ sóng của các TAs trong một TAI list thì nó bắt buộc phải thực hiện một công việc được gọi là TAU (Tracking Area Update – cập nhật khu vực theo dõi) - thông báo cho mạng rằng nó vừa di chuyển. Sau khi UE thực hiện TAU, nó sẽ được gán vào một TAI list mới. Lấy ví dụ trong Hình 3, ta có TA A và TA B nằm trong một TAI list,

UE có thể di chuyển giữa tất cả các Cell (ô) trong hai TA này mà không cần thực hiện bất kỳ cập nhật nào. Nhưng nếu UE di chuyển tới một Cell bất kỳ ở TA D, nó sẽ gửi một tin nhắn thông báo TAU tới một eNB bất kỳ thuộc TA D và nó sẽ được gán vào một TAI list mới, TAI list mới này có thể chứa TA B và TA D hoặc TA C và TA D.

Mạng lõi (CORE NETWORK)

Core Network có vai trò điều khiển các tín hiệu thành phần của mạng viễn thông. Một số tác vụ của CN có thể kể tới là chịu trách nhiệm định tuyến dữ liệu đến các khu vực khác nhau hoặc các công nghệ vô tuyến khác cũng như theo dõi các UE trong mạng. CN cho mạng 4G hay LTE được gọi là EPC.

Hình 4 thể hiện một phần của EPC và cách nó được kết nối với E-UTRAN. CN phục vụ cho nhiều các công nghệ vô tuyến khác như GSM và UMTS. Giao diện điều khiển được thiết lập giữa eNB và MME trên nền của kết nối S1 được gọi là S1-MME hoặc S1-C và S1-U là giao diện người dùng kết nối eNB với S-GW.

Ta có thể kể tới một số chức năng chính của MME như:

- Thiết lập tín hiệu NAS cho phép UE kết nối trực tiếp với MME mà không cần thông qua E-UTRAN như trong Hình 1.
- Quản lý các TAI list và các UE hiện hữu tại các TAI list đó.
- Lựa chọn S-GW phù hợp để phục vụ cho các UE.
- Bàn giao việc xử lý các tác vụ của UE cho MME mới khi UE tiến hành TAU sang một TA mới không được MME cũ quản lý.

Khái niệm và vai trò của gọi nhắn (Paging) trong mạng di động 4G

Khi nhắc tới khái niệm paging, đầu tiên ta sẽ tìm hiểu về hai trạng thái của một UE. Mỗi UE luôn nằm ở một trong hai trạng thái là: *Trạng thái nhàn rỗi (Idle mode)* và *Trạng thái kết nối (Connected mode)*. Trong đó:

- *Connected mode (chế độ được kết nối)* của UE trong mạng di động là trạng thái khi thiết bị tham gia hoạt động truyền dữ liệu và truyền thông với mạng một cách liên tục và liên tục. Trong trạng thái này, UE được kích hoạt và có kết nối liên tục với mạng để truyền và nhận dữ liệu, tham gia vào cuộc gọi thoại, video, truyền phát đa phương tiện và các dịch vụ mạng khác.
- *Idle mode (chế độ nghỉ)* của UE trong mạng di động là trạng thái khi thiết bị không hoạt động hoặc không tham gia vào các hoạt động truyền dữ liệu hoặc cuộc gọi. Trong trạng thái

này, UE không thực hiện việc truyền dữ liệu hoặc nhận dữ liệu từ mạng mà chỉ duy trì kết nối với mạng để tiếp nhận thông báo và tìm kiếm các tín hiệu từ cơ sở hạ tầng mạng. Trong Idle mode, UE tiêu thụ ít năng lượng hơn so với khi hoạt động và không gây tải cho mạng. Tuy nhiên, UE vẫn duy trì khả năng nhận các thông báo, cuộc gọi hoặc các yêu cầu từ mạng để có thể kích hoạt và chuyển sang trạng thái hoạt động khi cần thiết. Khi UE trong trạng thái Connected mode, thông tin về kết nối giữa UE và eNB (hay còn gọi là eNB UE context) sẽ được duy trì liên tục. eNB UE context bao gồm các thông tin quan trọng như địa chỉ IP của UE, thông tin xác thực, tài nguyên cấp phát, trạng thái kết nối, và các thông số khác liên quan đến kết nối của UE với eNB giúp Core Network định vị được vị trí chính xác của UE đến cấp độ Cell. Connected mode ít tiết kiệm năng lượng hơn Idle mode, và sử dụng nhiều tài nguyên mạng và tăng tín hiệu vì UE phải thực hiện việc chuyển cell mỗi khi nó thay đổi cell và việc duy trì liên tục UE context trong Connected mode khi không cần thiết cũng sẽ làm lãng phí tài nguyên mạng một cách vô ích. Khi UE ở Idle mode, nó không thể gửi hoặc nhận bất kỳ thông tin nào đến Core Network. Do đó, Idle mode yêu cầu lượng tín hiệu thấp và tài nguyên mạng tối thiểu. Nhược điểm của Idle mode là mất thời gian để thiết lập kết nối. Lý do là vì khi UE chuyển từ Connected mode sang Idle mode, eNB loại bỏ eNB UE context - thông tin cần thiết để UE duy trì kết nối, điều này có nghĩa là mỗi khi UE ở Idle mode cần được kết nối lại với mạng, nó phải thiết lập một eNB UE context mới bao gồm nhận thông tin chung từ EPC và thiết lập một kết nối bảo mật với UE. Hơn nữa, khi UE ở Idle mode, vị trí của UE được biết với độ chính xác theo TAI list và để duy trì kết nối, vị trí của mạng phải biết với độ chính xác theo mức Cell. Do đó, UE phải được paging (gọi nhắn) mỗi khi dữ liệu hoặc tín hiệu điều khiển được gửi đến UE. Vậy nói ngắn gọn, paging chính là việc định vị chính xác vị trí của UE trên cấp độ Cell khi UE chuyển từ Idle mode sang Connected mode.

Quá trình diễn ra tác vụ paging

Hình 5 thể hiện luồng tín hiệu khi cần paging một UE. Vị trí của mỗi UE trong Idle mode được MME quản lý theo TAI list. Luồng bản tin paging có thể thấy trong Hình 5. Đầu tiên, MME sẽ tiến hành paging tại eNB cuối cùng mà UE hiện hữu, một bộ đếm thời gian tại MME sẽ đếm kể từ khi tín hiệu paging được gửi đi. Nếu UE đã di chuyển khỏi eNB cũ, bộ đếm thời gian tại MME sẽ hết hạn do UE không phản hồi lại bản tin đó và MME có thể tiến hành paging toàn bộ eNB trong TAI list. Khi đang ở trong Idle mode, UE luôn lắng nghe tín hiệu paging theo một chu kỳ DRX (Discontinuous Reception) tại Cell mà nó đang kết nối. Khi nhận được tín hiệu paging, UE bắt đầu quy trình truy cập ngẫu nhiên và bắt đầu thiết lập kết nối RRC (Radio Resource Control - một lớp giao thức giữa UE và eNB có nhiều chức năng bao gồm cả thiết lập và giải phóng kết nối) với eNB mới. Sau khi kết nối RRC được thiết lập, eNB

gửi một tin nhắn khởi tạo UE cho MME trong đó bao gồm bản tin NAS, ID tham chiếu tín hiệu UE và thông tin về các giao diện kết nối S1 khác. MME sẽ tiếp tục gửi yêu cầu thiết lập eNB UE context ban đầu cho eNB, bao gồm ID MME UE S1, các khóa bảo mật và các thông tin liên quan khác. Sau đó, eNB sẽ trả về một phản hồi thiết lập.

Paging cũng có thể được thực hiện theo nhiều cách khác nhau. Ví dụ, thuật toán paging tại MME có thể quyết định sẽ paging một khu vực lớn ngay lập tức hoặc cũng có thể chia thành nhiều nhiều giai đoạn, đầu tiên paging một khu vực nhỏ và nếu không có kết quả thì sẽ paging một khu vực lớn hơn và tiếp tục như vậy. Lựa chọn đầu tiên trung bình sẽ có thời gian thực là ngắn hơn, nhưng lựa chọn thứ hai có thể có chi phí thấp hơn về tổng thể và nếu không có ràng buộc nghiêm ngặt về thời gian, cách này có thể được ưu tiên hơn. Ta có thể tối ưu tài nguyên của mạng dành cho tác vụ paging bằng cách dự đoán vị trí hoặc một vùng mà UE có thể hiện hữu bằng các thuật toán dựa trên học máy hoặc các phép tính khác từ đó giới hạn số Cell hoặc eNB mà MME cần gửi bản tin paging tới. Trong phạm vi của giải pháp hữu ích này, các thuật toán sẽ dự đoán theo eNB chứ không phải theo Cell.

Do đó, có nhu cầu cần một giải pháp cải tiến hơn tác vụ paging nhằm tăng hiệu quả bằng cách giảm thời gian và chi phí (overhead) tổng thể cho việc paging bằng cách dự đoán vị trí hoặc một vùng mà UE có thể hiện hữu theo cách chính xác hơn.

Bản chất kỹ thuật của giải pháp hữu ích

Mục đích của giải pháp là giảm thời gian và chi phí (overhead) tổng thể cho việc paging bằng cách dự đoán vị trí hoặc một vùng mà UE có thể hiện hữu theo cách chính xác hơn.

Theo một phương án, giải pháp hữu ích đề xuất phương pháp gọi nhắn (paging) thiết bị người dùng (UE) trong mạng truyền thông di động 4G dựa trên dự đoán vận tốc UE, phương pháp này bao gồm các bước:

B1: thu thập dữ liệu từ nhật ký hoạt động của các UE, dữ liệu này gồm ít nhất thời gian sự kiện thủ tục, trạng thái kết nối và các eNB đã đăng ký;

B2: đọc dữ liệu nêu trên và khởi tạo vận tốc ngưỡng V_{th} ;

B3: kiểm tra UE chưa từng xuất hiện trong mạng:

nếu đúng, xác định gửi tín hiệu gọi nhắn đến tất cả các TA (Tracking Area – khu vực theo dõi),

nếu sai, xếp UE vào một trong hai trạng thái: ổn định (stable) nghĩa là UE trong trạng thái bình thường, không xảy ra hiện tượng lỗi đường truyền; hoặc không ổn định (unstable)

nghĩa là UE bị rơi vào vùng đường truyền bị ảnh hưởng bởi các yếu tố ngoại cảnh hoặc trùng lặp tín hiệu không dây giữa hai eNB, dựa vào vận tốc giả định của UE, trong đó vận tốc giả định của UE được tính theo biểu thức (3) dưới đây:

$$d = 2 * r * \arcsin(a) \quad (1)$$

với

$$a = \sqrt{0.5 - \frac{\cos[(lat_2 - lat_1) * p]}{2} + \cos(lat_1 * p) * \cos(lat_2 * p) * \frac{1 - \cos[(lon_2 - lon_1) * p]}{2}} \quad (2)$$

trong đó:

$d(\text{km})$ là khoảng cách giữa hai điểm,

$r(\text{km})$ là bán kính trái đất,

lat_1 và lat_2 là vĩ độ của hai điểm,

lon_1 và lon_2 là kinh độ của hai điểm,

$$p = \pi/180$$

$$v = \frac{d(eNB_1, eNB_2)}{\Delta_t} \quad (3)$$

$$\Delta_t = \frac{Timestamp_2 - Timestamp_1}{1000} \quad (4)$$

trong đó:

$v(\text{km/s})$ là vận tốc giả định của UE,

$d(eNB_1, eNB_2)(\text{km})$ là khoảng cách giữa hai eNB bắt được tín hiệu của UE tại $Timestamp_1$ và $Timestamp_2$, được tính theo biểu thức (1)

$\Delta_t(\text{s})$ là khoảng thời gian giữa hai điểm dữ liệu, được tính theo biểu thức (4); và chuyển sang B4;

B4: kiểm tra điều kiện $0 < V(\text{Velocity}) < V_{th}$:

nếu đúng: xếp UE vào trạng thái ổn định (stable), thiết lập pagingSet(t_0) = nhóm (cluster) (eNB $t-1$), nghĩa là thiết lập tập các eNB nằm trong bán kính R của eNB ngay trước đó cho MME khi UE đang ở trạng thái ổn định (stable), và

nếu sai: nếu vận tốc giả định của UE = 0 (UE không thay đổi eNB) thì UE vẫn trong trạng thái ngay trước đó của nó, và xếp UE vào trạng thái không ổn định (unstable), thiết lập pagingSet(t_0) = nhóm (cluster) (eNB $t-1$) hoặc = pagingSet($t-1$), nghĩa là thiết lập tập các eNB nằm trong bán kính R của eNB ngay trước đó khi UE đang ở trạng thái ổn định (stable), và thêm cả các eNB đã gọi ý ngay trước đó kế tiếp giúp ít tốn phí tổn điều khiển (overhead) hơn khi paging;

đặt V_{th} lớn ($\geq 0,05$ km/s) để tránh UE sẽ rơi vào trạng thái unstable quá nhiều dẫn tới số lượng eNB cần paging cũng tăng theo mặc dù không cần thiết;

trong đó, phương pháp này còn khác biệt ở chỗ:

chỉ sử dụng thông tin vị trí của các eNB hay khoảng cách giữa các eNB mà đã kết nối liên tiếp với UE mà được lưu trong bộ nhớ của hệ thống truyền thông mà không yêu cầu UE cung cấp thông tin riêng tư về vị trí, tốc độ của UE.

Mô tả vắn tắt các hình vẽ

Giải pháp hữu ích sẽ được mô tả dựa vào các hình vẽ kèm theo trong đó:

Hình 1 thể hiện giao diện kết nối giữa EPC, E-UTRAN và UE;

Hình 2 thể hiện giao diện S1 giữa MME / Serving Gateway với eNB, X2 giữa các eNB;

Hình 3 thể hiện E-UTRAN được chia thành nhiều TAs, eNB và Cells;

Hình 4 thể hiện giao diện giữa MME, S-GW và E-UTRAN;

Hình 5 thể hiện luồng tín hiệu khi cần paging một UE;

Hình 6 thể hiện sơ đồ thuật toán số 1;

Hình 7 thể hiện sơ đồ thuật toán số 2;

Hình 8 thể hiện CPU trước và trong khi thuật toán thứ nhất vận hành;

Hình 9 thể hiện RAM trước và trong khi thuật toán thứ nhất vận hành;

Hình 10 thể hiện độ chính xác và độ chính xác thực tế với các giá trị R(km) và V_{th} (km/s) khác nhau của 2808 UE;

Hình 11 thể hiện độ chính xác và độ chính xác thực tế với các giá trị R(km) và Vth(km/s) khác nhau trên toàn bộ dữ liệu;

Hình 12 thể hiện số lượng eNB trung bình cần paging;

Hình 13 thể hiện CPU trước và trong khi thuật toán thứ hai vận hành;

Hình 14 thể hiện RAM trước và trong khi thuật toán thứ hai vận hành.

Mô tả chi tiết giải pháp hữu ích

Sau đây, giải pháp hữu ích sẽ được mô tả chi tiết theo các phương án ưu tiên thực hiện khác nhau dựa vào các hình vẽ kèm theo, cùng với các biểu thức, ký hiệu chỉ dẫn. Các chi tiết, thành phần giống nhau sẽ được biểu thị bằng các ký hiệu chỉ dẫn giống nhau. Các hình vẽ chỉ có mục đích minh họa chứ không nhằm giới hạn phạm vi của giải pháp.

Trước hết, phần này sẽ trình bày một thuật toán thử nghiệm đơn giản:

Thuật toán dựa trên các eNB nằm trong bán kính

Trong dữ liệu sử dụng, không có bất cứ thông tin nào về vị trí địa lý chính xác của UE hay các Cells mà chỉ có vĩ độ và kinh độ của các eNB. Đó là lý do vì sao các thuật toán là tìm kiếm danh sách các eNB tiềm năng chứ không phải các Cells. Đầu tiên, hãy thử nghiệm một thuật toán đơn giản là paging tất cả eNB trong bán kính R(km) của eNB trước đó (bao gồm cả chính eNB đó). Từ dữ liệu về kinh độ và vĩ độ của các eNB, có thể tính khoảng cách giữa 2 eNB bằng biểu thức 1.

$$d = 2 * r * \arcsin(a) \quad (1)$$

Với

$$a = \sqrt{0.5 - \frac{\cos[(lat_2 - lat_1) * p]}{2} + \cos(lat_1 * p) * \cos(lat_2 * p) * \frac{1 - \cos[(lon_2 - lon_1) * p]}{2}} \quad (2)$$

Trong đó:

- d(km) là khoảng cách giữa hai điểm.
- r(km) là bán kính trái đất.
- lat1 và lat2 là vĩ độ của hai điểm.
- lon1 và lon2 là kinh độ của hai điểm.

- $p = \pi/180$

Từ biểu thức 1 và dữ liệu về lat (vĩ độ - latitude) và lon (kinh độ - longitude) của các eNB, ta có thể liệt kê được một danh sách các eNB nằm trong bán kính R của từng eNB một, từ đó mỗi khi cần paging, chỉ cần gửi tín hiệu cho các eNB nằm trong danh sách được xác định từ trước. Ta có sơ đồ thuật toán như Hình 6, trong đó:

- t_0 : Timestamp tại thời điểm hiện tại.
- $t-k$: Timestamp tại thời điểm -k trong quá khứ, mỗi điểm dữ liệu trong quá khứ nhận được từ hệ thống tương ứng với tăng k lên 1.
- $\text{pagingSet}(t_0)$: Danh sách các eNB thuật toán gợi ý cho MME.
- $\text{cluster}(\text{eNB}(t-k))$: Danh sách các eNB nằm trong bán kính R của eNB tại thời điểm -k.

Hình 6 thể hiện sơ đồ thuật toán số 1. Thuật toán này khá đơn giản và cũng cho hiệu quả paging là khá tốt nếu UE không di chuyển nhiều và không có vấn đề gì trong quá trình truyền sóng giữa UE và các eNB. Nhưng thực tế không hề đơn giản như vậy, lỗi truyền sóng giữa UE và các eNB có thể dẫn tới trường hợp một eNB lại có thể bắt được tín hiệu từ một UE ở cách xa mà không liên quan gì tới nó, điều có thể giải quyết phần nào ở thuật toán tiếp theo.

Sau đây, một phương án ưu tiên thực hiện của giải pháp hữu ích sẽ được mô tả chi tiết.

Trước hết, phần sau đây sẽ mô tả về thuật toán paging được giải pháp hữu ích đề xuất. Thuật toán dựa trên vận tốc giả định của UE

Phân tích dữ liệu thu thập từ nhật ký hoạt động của các UEs, (ví dụ: thời gian sự kiện thủ tục, trạng thái kết nối và các eNB đã đăng ký), có thể quan sát được một số tình huống như sau: Khi UE đang di chuyển và sử dụng các dịch vụ trong mạng, lịch sử các eNB thay đổi thường xuyên, di chuyển và TAU xảy ra, UE trong Connected mode. Trong trường hợp như vậy, vị trí của các eNB nhận tín hiệu có thể không phản ánh vị trí "thực" của UE vì khoảng cách giữa 2 eNB là quá xa để UE đạt được trong một thời gian ngắn. Tính biến đổi của truyền thông không dây do đa đường phân tán, điều kiện thời tiết, môi trường triển khai, ..., mới chính là các nguyên nhân dẫn đến sự thay đổi bất hợp lý này. Do đó, sai số trong thủ tục paging sẽ rất lớn khi quyết định paging dựa trên các eNB xuất hiện trong trạng thái truyền thông không ổn định như vậy. Khi UE liên tục được bắt sóng bởi 2 eNB cách xa nhau trong thời gian ngắn, vấn đề xảy ra là giữa 2 eNB này thì MME không thể nhận biết được đâu mới chính là eNB

thực nơi UE hiện hữu để gửi yêu cầu paging hoặc bất cứ bản tin điều khiển nào khác khi cần thiết. Để giải quyết vấn đề trên, thuật toán này sẽ xếp UE vào một trong 2 trạng thái:

- Stable: UE trong trạng thái bình thường, không xảy ra hiện tượng lỗi đường truyền.

- Unstable: UE bị rơi vào vùng đường truyền bị ảnh hưởng bởi các yếu tố ngoại cảnh hoặc trùng lặp tín hiệu không dây giữa 2 eNB, lúc này có khả năng 2 hoặc nhiều eNB sẽ liên tục nhận được tín hiệu của UE. Để xác định được UE đang nằm trong trạng thái nào, cần tính được vận tốc của UE, do không có thông tin chính xác về vị trí của UE, sẽ coi vị trí của eNB khi nhận được tín hiệu chính là vị trí của UE. Vận tốc giả định của UE sẽ được tính theo biểu thức 3.

$$v = \frac{d(eNB_1, eNB_2)}{\Delta_t} \quad (3)$$

$$\Delta_t = \frac{Timestamp_2 - Timestamp_1}{1000} \quad (4)$$

trong đó:

- $v(\text{km/s})$: Vận tốc giả định của UE.
- $d(eNB_1, eNB_2)(\text{km})$: Khoảng cách giữa 2 eNB bắt được tín hiệu của UE tại Timestamp1 và Timestamp2, được tính theo biểu thức 1.
- $\Delta_t (s)$: Khoảng thời gian giữa 2 điểm dữ liệu, được tính theo biểu thức 4. Ta có sơ đồ thuật toán số 2 như Hình 7.

Trong đó:

- t_0 : Timestamp tại thời điểm hiện tại.
- $t-k$: Timestamp tại thời điểm -k trong quá khứ, mỗi điểm dữ liệu trong quá khứ nhận được từ hệ thống tương ứng với tăng k lên 1.
- $\text{pagingSet}(t_0)$: Danh sách các eNB thuật toán gợi ý cho MME.
- $\text{pagingSet}(t-k)$: Danh sách các eNB thuật toán gợi ý cho MME tại thời điểm -k.
- $\text{cluster}(eNB(t-k))$: Danh sách các eNB nằm trong bán kính R của eNB tại thời điểm -k.
- V_{th} : Một vận tốc ngưỡng để xác định trạng thái của UE.

- status: Trạng thái của UE (stable hoặc unstable).

Theo sơ đồ thuật toán trên, đầu tiên UE mặc định sẽ trong trạng thái stable, khi nhận được thông tin của UE từ hệ thống, có thể tính được vận tốc giả định (velocity) của UE khi UE di chuyển từ eNB(t-2) tới eNB(t-1). Lúc này có hai trường hợp xảy ra:

- Vận tốc giả định lớn hơn 0 và nhỏ hơn vận tốc ngưỡng (V_{th}): UE vẫn nằm trong trạng thái stable.
- Vận tốc giả định lớn hơn vận tốc ngưỡng (V_{th}): UE rơi vào trạng thái unstable.

Nếu vận tốc giả định của UE = 0 (UE không thay đổi eNB) thì UE vẫn trong trạng thái ngay trước đó của nó. UE chỉ chuyển từ unstable sang stable khi vận tốc giả định lớn hơn 0 và nhỏ hơn vận tốc ngưỡng, chuyển từ stable sang unstable khi vận tốc giả định lớn hơn vận tốc ngưỡng. Chiến thuật sẽ gợi ý tập các eNB nằm trong bán kính R của eNB ngay trước đó cho MME khi UE đang stable và sẽ gợi ý thêm cả các eNB đã gợi ý trước đó cho MME nếu UE rơi vào unstable. Số lượng eNB gợi ý sẽ ngày càng tăng lên nếu UE unstable qua quá nhiều eNB. Khả năng UE rơi vào unstable cũng thay đổi phụ thuộc vào vận tốc ngưỡng (V_{th}) mà ta cài đặt vào chiến thuật. Vì giả định vị trí của UE là vị trí của eNB nên vận tốc giả định của UE thường khá lớn, vậy nên cũng cần đặt V_{th} lớn (≥ 0.05 km/s) vì khi đặt V_{th} nhỏ UE sẽ rơi vào unstable quá nhiều dẫn tới số lượng eNB cần paging cũng tăng theo mặc dù không cần thiết.

Hiệu quả kỹ thuật đạt được của giải pháp hữu ích

Phần này sẽ trình bày hiệu quả tác vụ paging lần thứ nhất của các thuật toán đã giới thiệu ở chương trước.

Phương pháp đánh giá hiệu quả paging

Mỗi thuật toán cần đánh giá hai thông số là độ chính xác, tốc độ xử lý và tải tiêu thụ.

Độ chính xác

Bảng 1. Thống kê toàn bộ dữ liệu

	File 1	File 2	File 3	Tổng
Số điểm dữ liệu	874252	983218	996637	2854107
Số UE	71544	74155	75038	220737
Số lần thay đổi eNB	137245	158973	161767	457985
Số UE không thay đổi eNB	54113	54789	55389	164291
Số lần gọi paging	637887	715850	727507	2081244

Bảng 2. Thống kê các UE được chọn

File 1	File 1	File 2	File 3	Tổng
Số điểm dữ liệu	50157	59432	66740	176329
Số UE	840	936	1032	2808
Số lần thay đổi eNB	28311	31341	34926	94578
Số lần gọi paging	23465	29568	34027	87060

Mỗi thuật toán cần đánh giá độ chính xác trên 2808 UE được chọn trong Bảng 3.4 và trên toàn bộ 3 tập dữ liệu trong Bảng 2. Trong số 176329 điểm dữ liệu của 2808 UE được chọn, có 87060 lần mạng thực hiện tác vụ paging. Tuy nhiên chỉ tính độ chính xác của thuật toán trên 79675 điểm do những điểm còn lại là lần đầu tiên UE được mạng phát hiện và cần paging toàn bộ TA vì chưa có bất cứ thông tin nào để đưa vào các thuật toán. Tương tự, có 2081244 lần hệ thống gọi paging trên toàn bộ 3 file dữ liệu nhưng chỉ tính độ chính xác trên 1807378 điểm dữ liệu. Mỗi lần paging được coi là chính xác nếu eNB thực nằm trong danh sách các eNB tiềm năng mà thuật toán đề xuất cho hệ thống. Trên thực tế, mặc dù paging toàn bộ TA nhưng vẫn có xác suất khoảng 1.65% không có eNB nào bắt được tín hiệu phản hồi từ UE (vì nhiều lý do khác nhau) nên sau khi tính ra độ chính xác từ bộ dữ liệu trên, cần nhân con số đó với một tỷ lệ là 98.35% để ra được kết quả cuối cùng sát nhất với thực tế.

Lưu ý: Khi doanh nghiệp vận hành hệ thống, cần paging nhiều lần tới khi tìm được vị trí của UE. Tuy nhiên, các kết quả trong chương này chỉ là xác suất paging chính xác lần đầu tiên.

Tốc độ xử lý và tải tiêu thụ

Toàn bộ quá trình thử nghiệm các thuật toán đều được thực hiện trên máy tính cá nhân với bộ dữ liệu ngoại tuyến nên các kết quả trong quá trình đánh giá đều mang tính chủ quan và có thể không giống với thực tế khi mà khối lượng dữ liệu trực tuyến cùng số lượng UE là rất lớn. Dưới đây là các thông số của máy tính cá nhân được sử dụng để thử nghiệm thuật toán:

- CPU: Intel(R) Xeon(R) CPU E5-2680 v4 @ 2.40GHz 14 Cores 28 threads.
- RAM: RAM ECC DDR4 16GB 2133MHz.
- Hệ điều hành: Windows 10 Pro 64-bit.
- Python: 3.10.9

Để đạt được tốc độ tối đa, các thuật toán đều được lập trình duyệt tuần tự qua các điểm sau khi chuyển toàn bộ dữ liệu từ dạng bảng sang dạng mảng hai chiều và sử dụng các hàm trong thư viện Numpy để xử lý.

Thuật toán thứ nhất

Thuật toán thứ nhất chỉ đề xuất *paging* các eNB nằm trong bán kính R (cluster) của eNB mới nhất bắt được tín hiệu từ UE. Bảng 3 thể hiện số lượng eNB trung bình mỗi cluster phụ thuộc vào bán kính R .

Bảng 3 Số lượng eNB trung bình mỗi cụm (cluster)

$R(km)$	Số lượng eNB trung bình
3	2.7
5	5.7
8	11.2
10	15.1

Bảng 4 và Bảng 5 lần lượt độ chính xác của thuật toán thứ nhất trên 2808 UE được chọn để phát triển thuật toán và toàn bộ tập dữ liệu với số lượng eNB cần paging trung bình ở cột eNB/paging. Độ chính xác thực tế là kết quả sau khi nhân số liệu ở cột Độ chính xác với 98.35% (đã trình bày ở phần trước). Độ chính xác của toàn bộ tập dữ liệu là khá lớn do đa số các UE không di chuyển hoặc di chuyển rất ít.

Bảng 4 Hiệu quả *paging* của thuật toán thứ nhất trên 2808 UE được chọn

$R(km)$	eNB/paging	Độ chính xác(%)	Độ chính xác thực tế(%)
3	4.15	89.76	88.28
5	8.17	95.48	93.9
8	14.96	97.74	96.13
10	19.23	98.19	96.57

Bảng 5 Hiệu quả *paging* của thuật toán thứ nhất trên toàn bộ tập dữ liệu

$R(km)$	eNB/paging	Độ chính xác(%)	Độ chính xác thực tế(%)
---------	------------	-----------------	-------------------------

3	2.97	96.84	95.24
5	6.06	98.68	97.05
8	11.8	99.43	97.79
10	15.65	99.59	97.94

Tốc độ xử lý và tải tiêu thụ

Dưới đây là thống kê tốc độ xử lý trung bình của thuật toán thứ nhất:

- Các UE được chọn:

- Thời gian xử lý một điểm dữ liệu: 31.4 μ s

- Khối lượng dữ liệu xử lý được trong 1s: 31847 điểm

- Toàn bộ tập dữ liệu:

- Thời gian xử lý một điểm dữ liệu: 26.3 μ s

- Khối lượng dữ liệu xử lý được trong 1s: 37996 điểm

Tốc độ xử lý trung bình của toàn bộ dữ liệu lớn hơn có thể lý giải là do nhiều UE có ít dữ liệu dẫn tới việc xử lý dữ liệu liên quan đến những UE này sẽ nhanh hơn. Do đó, tốc độ xử lý trung bình sẽ tăng lên. Hình 8 và Hình 9 lần lượt là chiếm dụng CPU và RAM trong thời gian vận hành của thuật toán thứ nhất. Hình 8 thể hiện CPU trước và trong khi thuật toán thứ nhất vận hành. Hình 9 thể hiện RAM trước và trong khi thuật toán thứ nhất vận hành.

Thuật toán thứ hai

Hiệu quả paging

Hình 10 thể hiện độ chính xác và độ chính xác thực tế với các giá trị R(km) và Vth(km/s) khác nhau của 2808 UE. Hình 11 Độ chính xác và độ chính xác thực tế với các giá trị R(km) và Vth(km/s) khác nhau trên toàn bộ dữ liệu.

Cụ thể hơn, Hình 10 và Hình 11 lần lượt là tỷ lệ paging chính xác cùng độ chính xác thực tế của thuật toán thứ hai trên 2808 UE và toàn bộ tập dữ liệu với các giá trị R và Vth khác nhau. Hình 12 cho thấy số lượng eNB trung bình cần paging trung bình của 2 bộ dữ liệu.

Tốc độ xử lý và tải tiêu thụ

Dưới đây là thống kê tốc độ xử lý trung bình của thuật toán thứ hai:

- 2808 UE :

- Thời gian xử lý một điểm dữ liệu: 277.73 μ s

- Khối lượng dữ liệu xử lý được trong 1s: 3600 điểm

- Toàn bộ tập dữ liệu:

- Thời gian xử lý một điểm dữ liệu: 254.84 μ s

- Khối lượng dữ liệu xử lý được trong 1s: 3924 điểm Hình 13 và Hình 14 lần lượt là chiếm dụng CPU và RAM trong thời gian vận hành của thuật toán thứ hai. Cụ thể hơn, Hình 13 thể hiện CPU trước và trong khi thuật toán thứ hai vận hành, và Hình 14 thể hiện RAM trước và trong khi thuật toán thứ hai vận hành.

YÊU CẦU BẢO HỘ

1. Phương pháp gọi nhắn (paging) thiết bị người dùng (UE) trong mạng truyền thông di động 4G dựa trên dự đoán vận tốc UE, phương pháp này bao gồm các bước:

bước 1: thu thập dữ liệu từ nhật ký hoạt động của các UE, dữ liệu này gồm ít nhất thời gian sự kiện thủ tục, trạng thái kết nối và các eNB đã đăng ký;

bước 2: đọc dữ liệu nêu trên và khởi tạo vận tốc ngưỡng V_{th} ;

bước 3: kiểm tra UE chưa từng xuất hiện trong mạng:

nếu đúng, xác định gửi tín hiệu gọi nhắn đến tất cả các TA (Tracking Area – khu vực theo dõi),

nếu sai, xếp UE vào một trong hai trạng thái: ổn định (stable) nghĩa là UE trong trạng thái bình thường, không xảy ra hiện tượng lỗi đường truyền; hoặc không ổn định (unstable) nghĩa là UE bị rơi vào vùng đường truyền bị ảnh hưởng bởi các yếu tố ngoại cảnh hoặc trùng lặp tín hiệu không dây giữa hai eNB, dựa vào vận tốc giả định của UE, trong đó vận tốc giả định của UE được tính theo biểu thức dưới đây:

$$v = \frac{d(eNB_1, eNB_2)}{\Delta_t}$$

trong đó:

v (km/s) là vận tốc giả định của UE,

Δ_t (s) là khoảng thời gian giữa hai điểm dữ liệu, được tính theo biểu thức:

$$\Delta_t = \frac{Timestamp_2 - Timestamp_1}{1000}$$

$d(eNB_1, eNB_2)$ (km) là khoảng cách giữa hai eNB bắt được tín hiệu của UE tại $Timestamp_1$ và $Timestamp_2$, được tính theo biểu thức:

$$d = 2 * r * \arcsin(a)$$

với

$$a = \sqrt{0.5 - \frac{\cos[(lat_2 - lat_1) * p]}{2} + \cos(lat_1 * p) * \cos(lat_2 * p) * \frac{1 - \cos[(lon_2 - lon_1) * p]}{2}}$$

trong đó:

$d(\text{km})$ là khoảng cách giữa hai điểm,

$r(\text{km})$ là bán kính trái đất,

$lat1$ và $lat2$ là vĩ độ của hai điểm,

$lon1$ và $lon2$ là kinh độ của hai điểm,

$$p = \pi/180;$$

bước 4: kiểm tra điều kiện $0 < V$ (Velocity) $< V_{th}$:

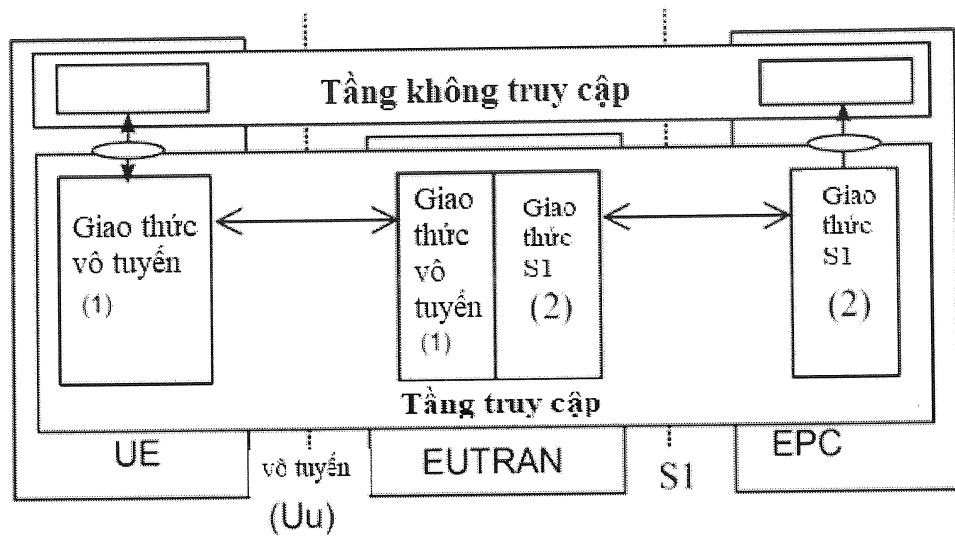
nếu đúng: xếp UE vào trạng thái ổn định (stable), thiết lập $\text{pagingSet}(t_0) = \text{nhóm (cluster) (eNB } t-1)$, nghĩa là thiết lập tập các eNB nằm trong bán kính R của eNB ngay trước đó cho MME khi UE đang ở trạng thái ổn định (stable), và

nếu sai: nếu vận tốc giả định của UE = 0 (UE không thay đổi eNB) thì UE vẫn trong trạng thái ngay trước đó của nó, và xếp UE vào trạng thái không ổn định (unstable), thiết lập $\text{pagingSet}(t_0) = \text{nhóm (cluster) (eNB } t-1)$ hoặc $= \text{pagingSet}(t-1)$, nghĩa là thiết lập tập các eNB nằm trong bán kính R của eNB ngay trước đó khi UE đang ở trạng thái ổn định (stable), và thêm cả các eNB đã gợi ý ngay trước đó kế tiếp giúp ít tốn phí tổn điều khiển (overhead) hơn khi paging;

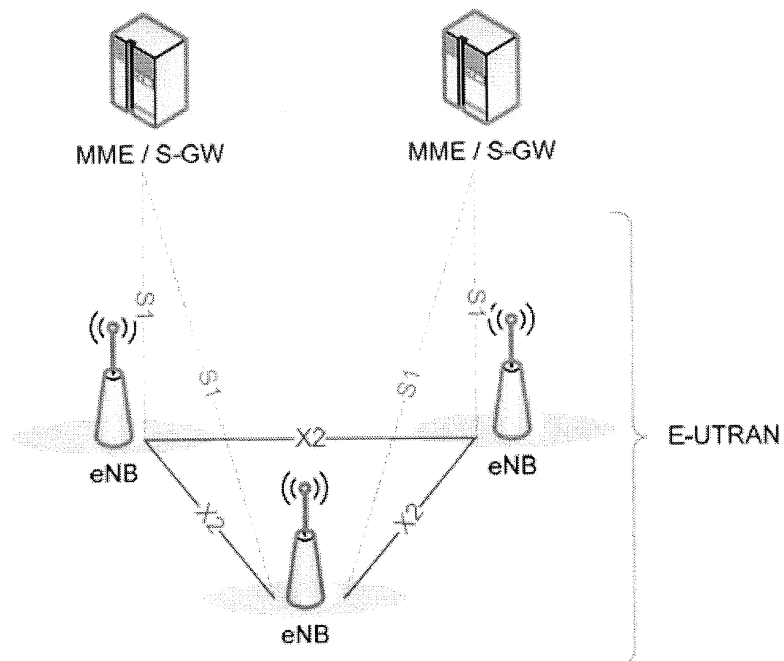
đặt V_{th} lớn ($\geq 0,05 \text{ km/s}$) để tránh UE sẽ rơi vào trạng thái unstable quá nhiều dẫn tới số lượng eNB cần paging cũng tăng theo mặc dù không cần thiết;

trong đó, phương pháp này còn khác biệt ở chỗ:

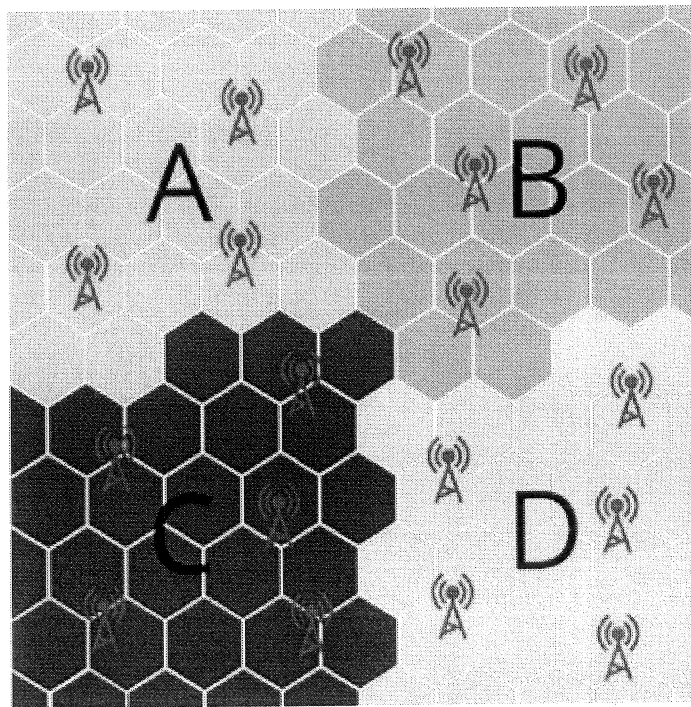
chỉ sử dụng thông tin vị trí của các eNB hay khoảng cách giữa các eNB mà đã kết nối liên tiếp với UE mà được lưu trong bộ nhớ của hệ thống truyền thông mà không yêu cầu UE cung cấp thông tin riêng tư về vị trí, tốc độ của UE.



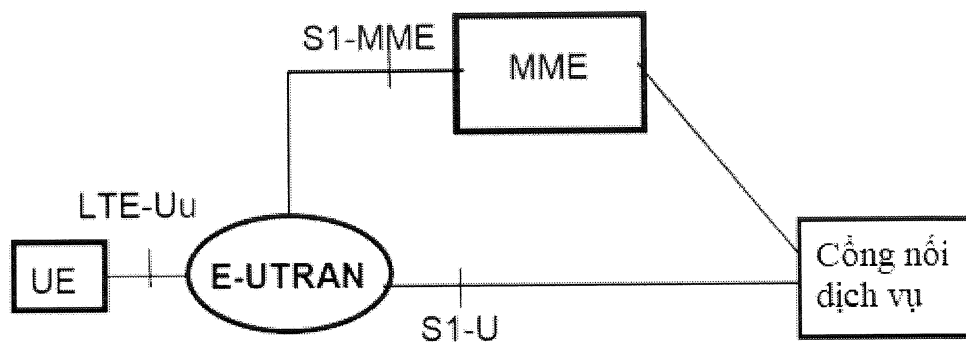
Hình 1



Hình 2

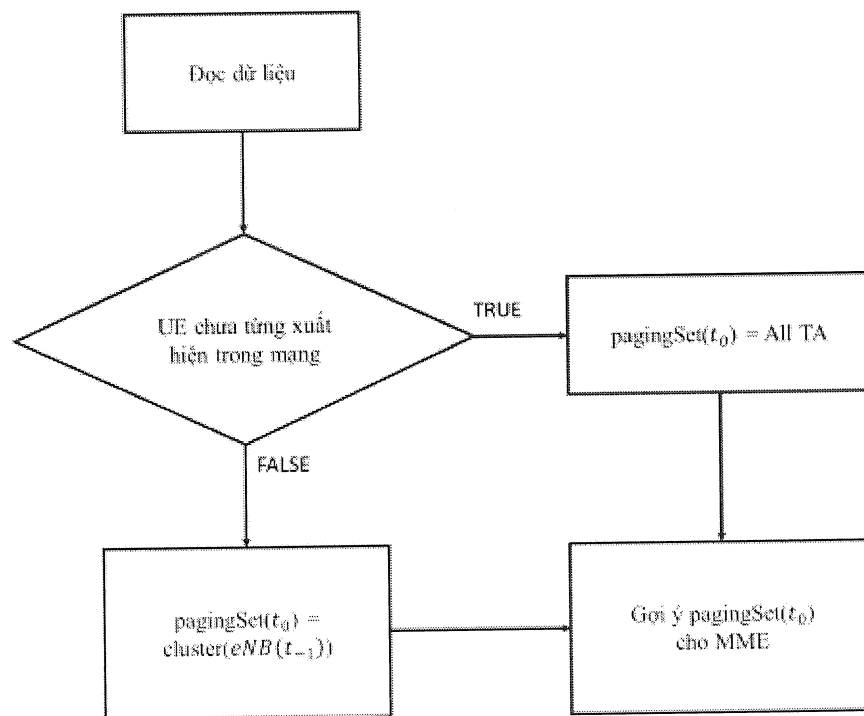


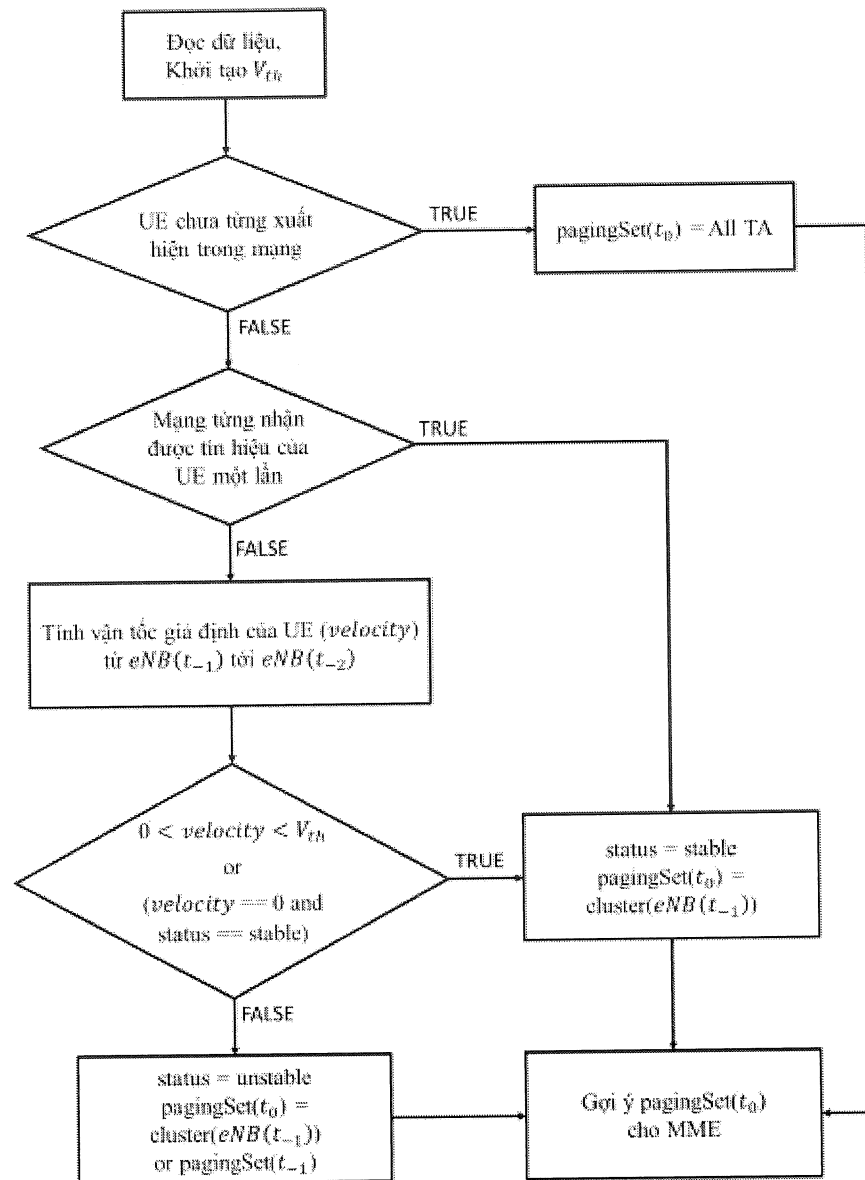
Hình 3



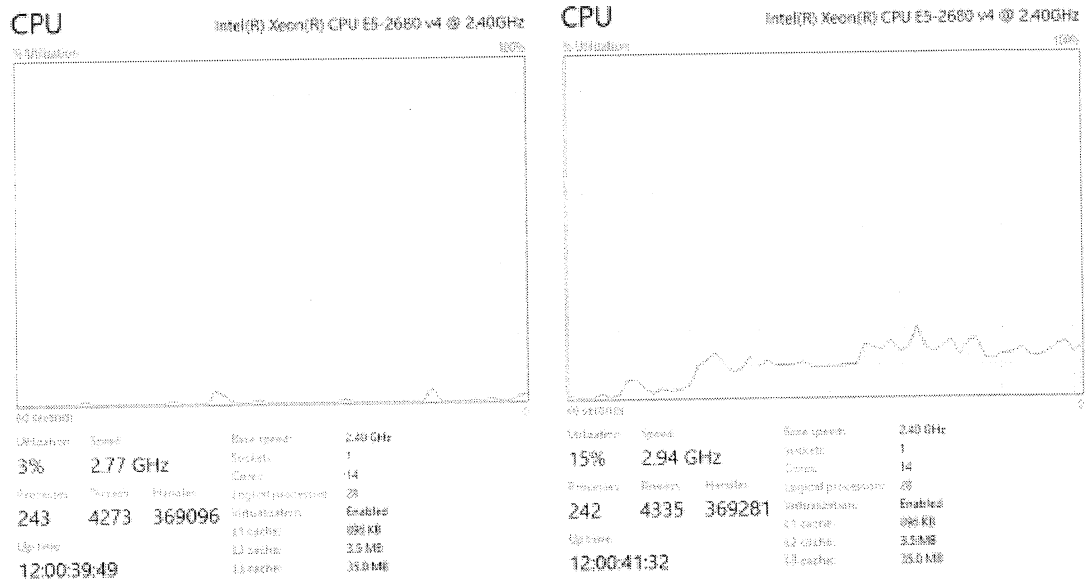
Hình 4



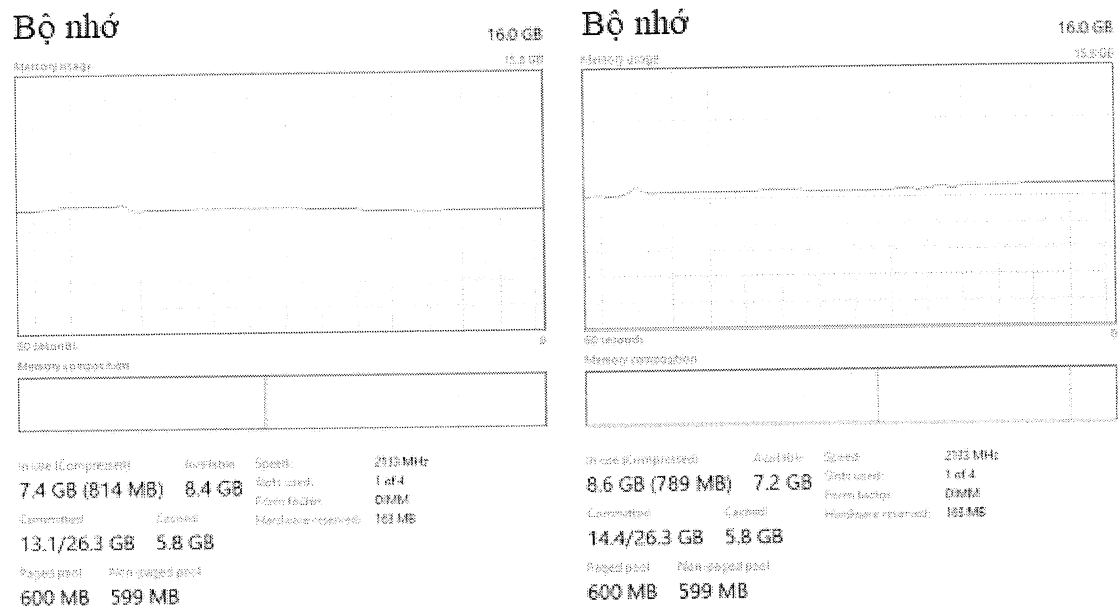
**Hình 6**



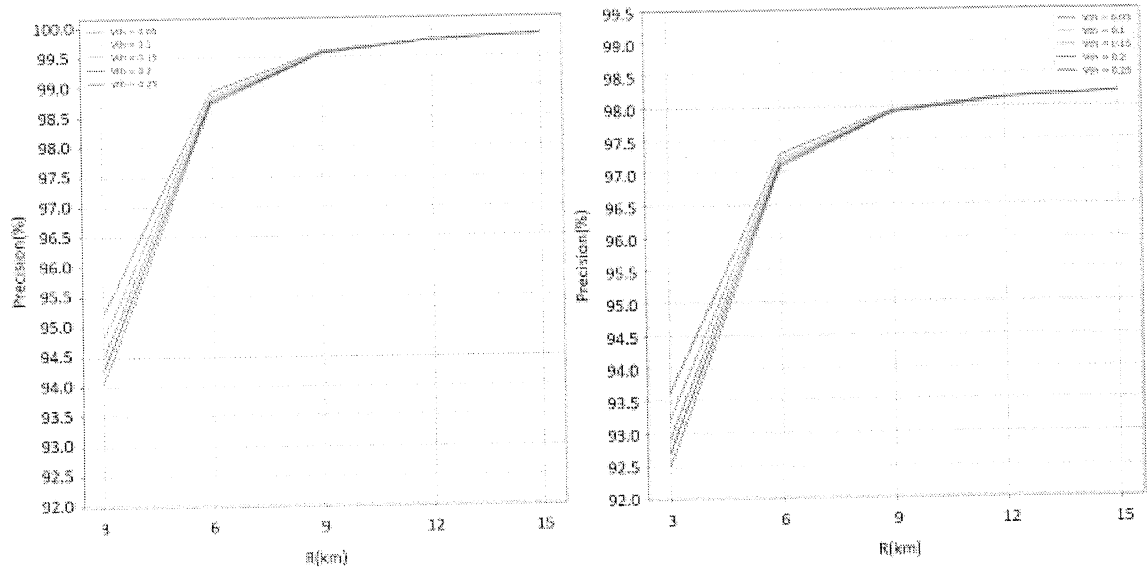
Hình 7



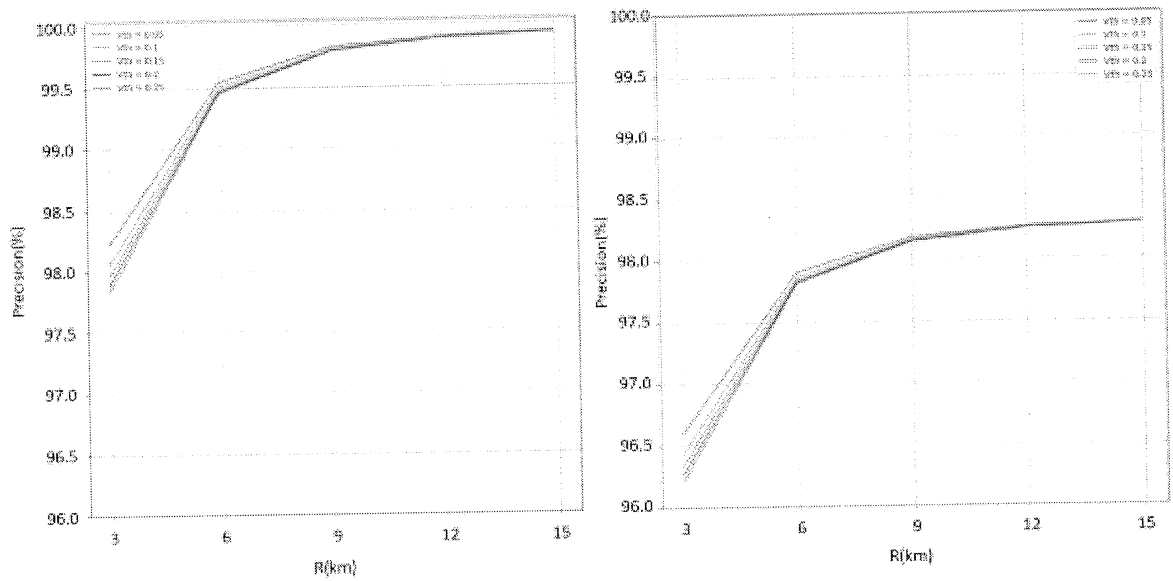
Hình 8



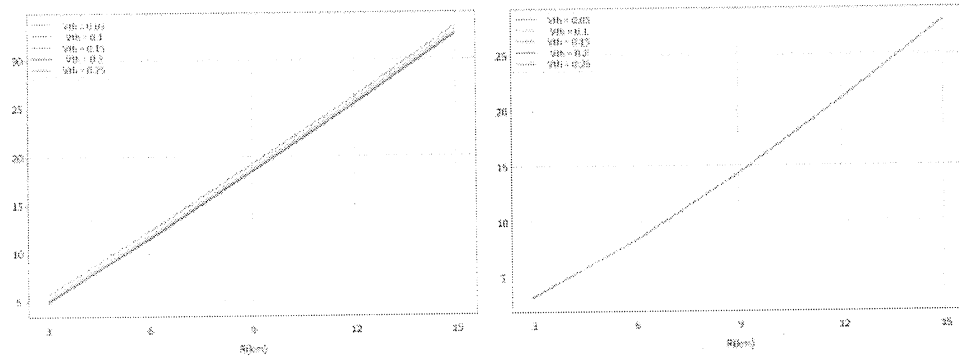
Hình 9



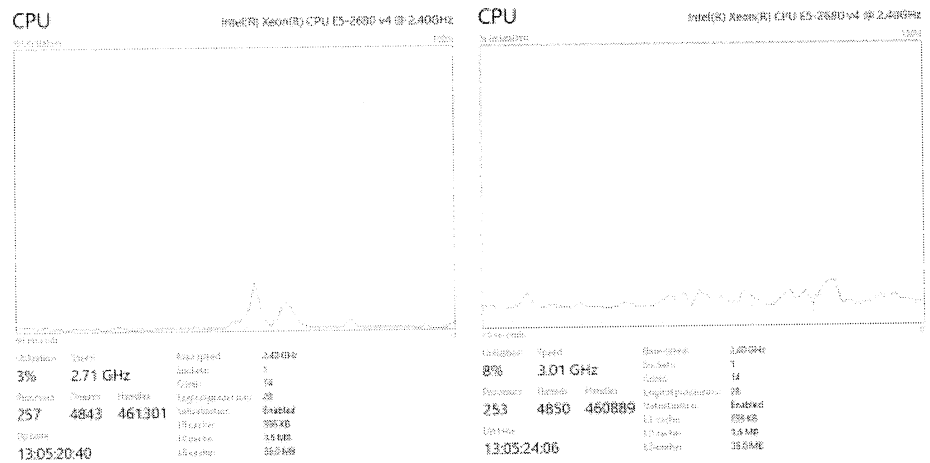
Hình 10



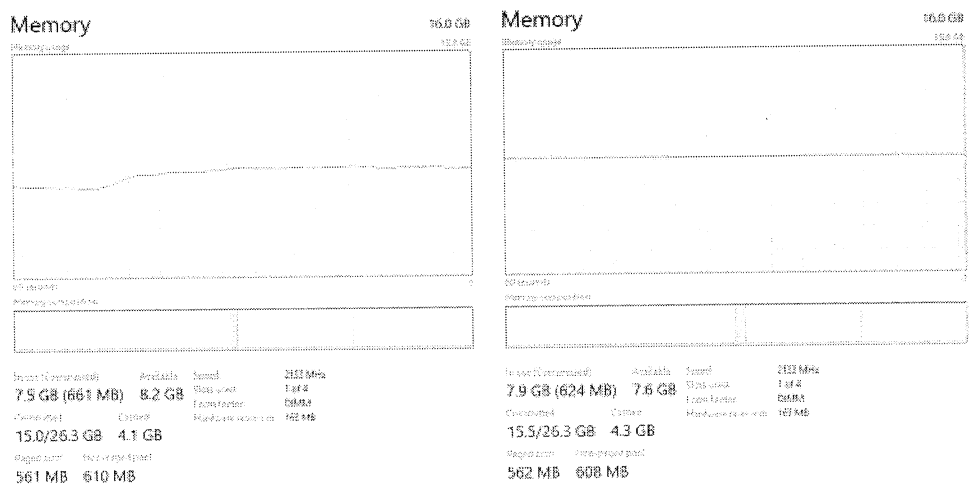
Hình 11



Hình 12



Hình 13



Hình 14