

# PHÁT TRIỂN THUẬT TOÁN GIÁM SÁT VÀ CẢNH BÁO MẬT ĐỘ XE TẠI CÁC HƯỚNG CỦA NGÃ TƯ DÙNG GPS TRÊN ĐIỆN THOẠI ANDROID

Huỳnh Hoàng Hà

Trường Đại học Sư phạm Kỹ thuật TP.HCM

Email: hahh@hcmute.edu.vn

Ngày nhận bài: 11/5/2021; Ngày chấp nhận đăng: 21/7/2021

## TÓM TẮT

Hiện nay, tình trạng kẹt xe xảy ra khá nghiêm trọng trong giờ cao điểm tại các giao lộ, khu công nghiệp, đoạn đường đang sửa chữa hay do va chạm trong lưu thông, v.v. Vì vậy, trong bài báo này, chúng tôi phát triển giải thuật theo dõi mật độ xe lưu thông theo các hướng tại các điểm ngã tư và cảnh báo tài xế khi có điểm kẹt xe phía trước. Chúng tôi sử dụng hệ thống định vị toàn cầu GPS (global positioning system) trên các thiết bị chạy Android. Các thiết bị này được gắn lên xe lưu thông hoặc mang theo bên người của tài xế (điện thoại Android), từ đó tính toán và đồng bộ số lượng phương tiện tại các hướng của ngã tư lên Firebase. Chương trình ứng dụng được cài đặt thuật toán xử lý, ứng dụng có khả năng thêm bớt được các giao lộ cần theo dõi, xác định được số lượng xe tại các hướng của các ngã tư, tính toán được khoảng cách giữa xe với các ngã tư và gửi cảnh báo cho tài xế khi có điểm đông đúc. Mật độ xe được xác định chính xác mà không phụ thuộc các yếu tố môi trường như trời tối, trời mưa, v.v.

*Từ khóa:* GPS, Android, Firebase, kẹt xe, giao thông.

## 1. GIỚI THIỆU VỀ ĐỀ TÀI

Vấn đề giao thông, kẹt xe là một vấn đề hết sức quan trọng ngày nay, đặc biệt là ở các thành phố lớn như Hồ Chí Minh, Hà Nội. Dân số đang tăng nhanh, cộng thêm việc tập trung các khu công nghiệp quá nhiều về trung tâm thành phố, việc ùn tắc giao thông là điều không thể tránh khỏi. Nhà nước có nhiều phương án được áp dụng để hạn chế việc tắc nghẽn giao thông tại các khu vực đông đúc, giờ cao điểm như: mở rộng đường, mở thêm các đường nhánh khác để giao thông có thể chia ra thành nhiều hướng di chuyển, đóng phí đường bộ, khuyến khích và sử dụng nhiều phương tiện công cộng hơn, v.v... Song song đó, các nhà nghiên cứu, tác giả khắp nơi trên thế giới cũng chung sức nghiên cứu và đề xuất các giải pháp để giảm bớt tình trạng kẹt xe như hiện nay.

Sun Ye từ trường đại học Shandong Jiaotong, Trung Quốc đề xuất giải pháp đóng phí tham gia giao thông để lượng xe lưu thông giảm, dẫn đến việc ùn tắc kẹt xe sẽ giảm. Mức phí áp dụng cho từng đoạn đường phụ thuộc vào các đặc điểm của đoạn đường đó như mục tiêu, chi phí, phạm vi, phương pháp cũng như tình trạng đông đúc của phương tiện lưu thông trên đoạn đường đó [1].

Năm 2012, Vipin Jain và Ashlesh Sharma phân tích về các tổn thất có thể xảy ra khi bị ùn tắc giao thông như tổn nhiên liệu, thời gian, tiền tệ, chậm trễ trong đi lại của người tham gia giao thông. Nguyên nhân gây ùn tắc là do mạng lưới giao thông quy hoạch chưa tốt, đường xá xung quanh các điểm nóng chưa được triển khai tốt. Từ đó, nghiên cứu đề xuất giải pháp là tận dụng các camera gắn sẵn ở các giao lộ để thu thập hình ảnh, nhận dạng và xác định số

lượng xe tại các điểm nóng nhằm điều hướng xe lưu thông, giảm bớt sự tắc nghẽn cục bộ [2].

Trong bài báo của Wadud và cộng sự, nghiên cứu về hệ thống giao thông thông minh được nghiên cứu dựa trên việc phát hiện phương tiện và phân loại các tuyến đường theo mật độ xe. Hệ thống mà bài báo phát triển nhằm giúp tài xế xác định lộ trình ít kẹt xe nhất để di chuyển cho phù hợp. Các kỹ thuật xử lý ảnh được trích liên tục từ camera ở các điểm kẹt xe được áp dụng nhằm xác định mật độ xe, tốc độ di chuyển của phương tiện. Sau đó, kết quả xử lý được cập nhật lên ứng dụng Android giúp tài xế dễ dàng lựa chọn tuyến đường đi ít kẹt xe nhất [3].

Năm 2017, Chandana & cộng sự ứng dụng các công nghệ trong hệ thống Internet of Things (IoT) vào việc xác định số lượng xe lưu thông qua các khu vực cần theo dõi. Bộ phát sóng RF được gắn trên các phương tiện tham gia giao thông nhằm phát hiện được xe khi chạy qua bộ thu thập sóng RF, từ đó đếm số lượng xe di chuyển sang khu vực cần theo dõi [4].

Nghiên cứu của nhóm tác giả Ayesha Atta và cộng sự ứng dụng công nghệ RFID vào nhận dạng phương tiện tham gia giao thông. Bằng cách gắn các thẻ từ RFID trên phương tiện, khi các phương tiện đi tới giao lộ, nơi đặt máy quét RFID, thì máy trung tâm sẽ nhận dạng được phương tiện đi qua, từ đó tổng kết được mật độ xe tại khu vực cần theo dõi [5].

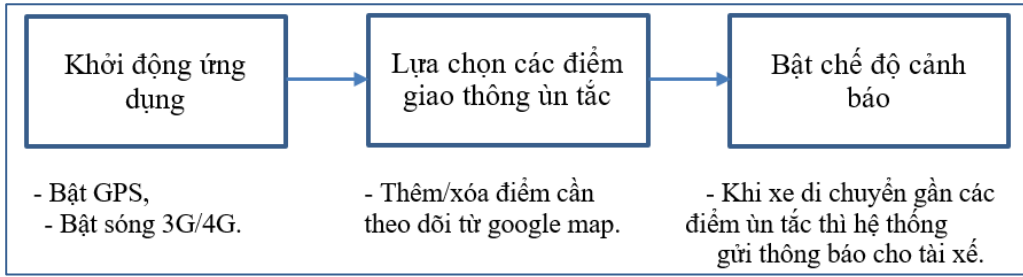
Trong công trình nghiên cứu của các tác giả Sandor Dornbush, Anupam Joshi ứng dụng hệ thống định vị toàn cầu GPS vào việc xác định phương tiện giao thông tại các điểm lắp đặt. Phương tiện giao thông được gắn các thiết bị phát GPS. Khi phương tiện chạy vào các giao lộ cần đo lường thì thiết bị GPS tiên hành kết nối với một bộ phát wifi và gửi tọa độ về một máy tính trung tâm để xử lý và xác định mật độ xe cũng như ước tính tốc độ xe di chuyển qua khu vực cần phân tích [6].

Các công trình nghiên cứu [2, 3] sử dụng Camera ở các giao lộ để nhận diện số lượng xe, tuy nhiên các phương pháp này khó thực hiện khi môi trường có nhiễu: trời mưa, trời tối, v.v và cần phải có 1 máy tính chủ cấu hình mạnh mới xử lý được dữ liệu hình ảnh. Các công trình số [4, 5] sử dụng công nghệ sóng không dây RF, RFID thì tránh được tình trạng ảnh hưởng của môi trường, nhưng việc xây dựng các trụ thu thập sóng tại tất cả các tuyến đường sẽ tốn rất nhiều chi phí. Công trình [6] tận dụng GPS có sẵn trên thiết bị di động chạy Android, các module có sẵn trên xe hơi, nên vừa có thể định vị chính xác số lượng xe và các trụ thu sóng nhưng công trình chỉ dừng lại nghiên cứu thuật toán định vị và chưa phát triển ứng dụng hỗ trợ tài xế.

Vì vậy, sau khi phân tích nhiều bài báo, chúng tôi quyết định nghiên cứu xây dựng một hệ thống tận dụng ưu điểm của các phương pháp trước đây, và xây dựng nên ứng dụng chạy trên các thiết bị Android được mang theo bên người tài xế. Hệ thống không sử dụng camera và các tài xế cần cài đặt phần mềm chạy ngầm để định vị vị trí của tài xế khi tham gia giao thông. Ứng dụng có các đặc điểm chính sau:

- Sử dụng GPS để định vị phương tiện tham gia giao thông.
- Số lượng phương tiện được tổng hợp và theo dõi tại Web Server có tên là Firebase.
- Phát triển ứng dụng quản lý các điểm, khu vực có giao thông phức tạp.
- Theo dõi và cảnh báo cho tài xế khi có các điểm nóng, kẹt xe xảy ra gần lộ trình di chuyển. Từ đó, tài xế có thể xem bản đồ để lựa chọn tuyến đường đi khác cho phù hợp.
- Ứng dụng được xây dựng hoàn toàn trên các thiết bị di động chạy Android có sẵn, hoặc các khối phần cứng chạy hệ điều hành Android trên xe hơi (trong nghiên cứu này, chúng tôi tiến hành chạy thử nghiệm trên thiết bị di động chạy Android).

Quy trình xử lý phần mềm ứng dụng được trình bày theo các bước như trong Hình 1.



Hình 1. Sơ đồ khởi của quy trình xử lý phần mềm ứng dụng

Với mục tiêu đặt ra là xây dựng hệ thống thu thập tọa độ từ phương tiện tham gia giao thông, phát triển ứng dụng dựa trên nền tảng thiết bị chạy Android, chúng tôi tiến hành phân tích vấn đề và chia quy trình nghiên cứu trong bài báo này thành 5 phần. Phần 1 giới thiệu tình hình nghiên cứu và hiện trạng trên thế giới cũng như ở Việt Nam. Phần 2 là quá trình phân tích, xây dựng thuật toán xử lý, xây dựng mô hình hệ thống. Phần 3 trình bày giao diện và tính năng của phần mềm ứng dụng. Phần 4 thử nghiệm thực tế với các thiết bị Android khi đi qua các điểm giao thông cần theo dõi, từ đó đánh giá độ chính xác cũng như khả năng báo động của hệ thống. Cuối cùng, phần 5 kết luận và đề xuất hướng phát triển tiếp theo của nghiên cứu.

## 2. XÂY DỰNG HỆ THỐNG

### 2.1. Mô hình hệ thống

Các phương tiện tham gia giao thông được trang bị một thiết bị chạy hệ điều hành Android. Trên thiết bị này, chúng tôi phát triển ứng dụng cập nhật tọa độ GPS của xe lên Web Server của điện thoại và ứng dụng có khả năng kiểm soát được số lượng xe tại khu vực cần theo dõi, tiến hành báo động cho tài xế khi phát hiện khu vực này có khả năng kẹt xe. Mô hình hệ thống được mô tả như Hình 2.

Ứng dụng Android cập nhật dữ liệu với Server thông qua sóng 3G/4G của chính thiết bị Android.



Hình 2. Mô hình hệ thống

### 2.2. Firebase

Dữ liệu tọa độ GPS thu thập từ các thiết bị Android cần được tổ hợp lại một trung tâm để tính toán số lượng phương tiện xe tại từng khu vực theo dõi. Bên cạnh Firebase, chúng ta có thể sử dụng các hệ thống thu thập dữ liệu khác như: Web Socket, Website HTTP. Mỗi loại hệ thống đều có những ưu điểm và nhược điểm riêng, được mô tả ngắn gọn trong Bảng 1.

Bảng 1. So sánh 1 số loại máy chủ

Website HTTP	Web Socket	Web Server (Firebase)
Chi phí thấp	Chi phí cao	Chi phí theo dung lượng sử dụng
Không có Real-Time	Có Real-Time	Có Real-Time*
Chi phí xây dựng hệ thống lớn	Chi phí xây dựng hệ thống lớn	Được Google xây dựng sẵn

\*Real-Time: Hệ thống có khả năng gửi thông báo tức thì, ngay khi có sự kiện xảy ra.

Dựa trên Bảng 1, Website HTTP có chi phí thấp nhất, nhưng không có Real-Time. Web Socket giúp cho hệ thống tối ưu tốt nhất về tính năng, nhưng chi phí xây dựng lớn, tốn nhiều thời gian. Vì vậy, trong nghiên cứu này, chúng tôi sử dụng Firebase, một Web Server được xây dựng sẵn bởi Google. Firebase có chi phí phụ thuộc vào dung lượng sử dụng, thời gian lập trình nhanh nên việc thử nghiệm là phù hợp nhất.

**2.3. Thuật toán tính khoảng cách trong tọa độ cầu**

$$a = \sin^2(\Delta\phi/2) + \cos \phi_1 * \cos \phi_2 * \sin^2(\Delta\lambda/2) \quad (1)$$

$$c = 2 * \text{atan2}( \sqrt{a}, \sqrt{(1 - a)} ) \quad (2)$$

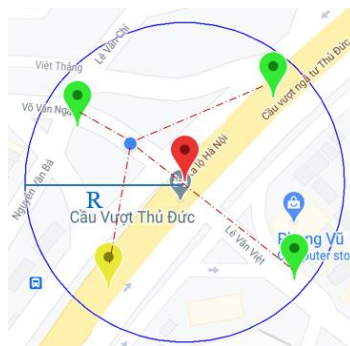
$$d = R * c \quad (3)$$

Bộ 3 công thức (1), (2), (3) [12] được sử dụng để tính toán khoảng cách giữa 2 tọa độ điểm trên bề mặt Trái đất. Với a là bình phương của nửa độ dài giữa 2 điểm, c là khoảng cách góc tính bằng radian, d là khoảng cách giữa 2 điểm tính bằng đơn vị là mét. Giả định gọi 2 điểm A & B có tọa độ là A(lat1, lon1), B(lat2, lon2), với lat là kinh độ, lon là vĩ độ. Khi đó, các thông số được tính toán như sau:

- R = 6371.3 km (bán kính Trái Đất)
- $\phi_1 = \text{lat1} * \pi/180;$
- $\phi_2 = \text{lat2} * \pi/180;$
- $\Delta\phi = (\text{lat2}-\text{lat1}) * \pi/180;$
- $\Delta\lambda = (\text{lon2}-\text{lon1}) * \pi/180;$

Dựa trên công thức (1)(2)(3), chúng tôi xác định được những phương tiện tham gia giao thông đang kẹt tại các điểm giao lộ cần theo dõi.

**2.4. Giải thuật xác định mật độ xe ở 4 hướng tại một ngã tư**



Hình 3. Tính toán vị trí xe đang thuộc hướng di chuyển nào khi vào giao lộ

Tại 1 điểm ngã tư, chúng tôi lấy tọa độ của 5 điểm: 1 điểm trung tâm và 4 điểm tại 4 hướng đường đi (Hình 3). Dựa trên công thức tính khoảng cách (1), (2), (3), chúng tôi đưa ra giải thuật xác định xe hiện tại ● đang thuộc hướng đi chuyển nào, như sau:

B1: Nếu xe vào trong bán kính R(100m) của giao lộ:

- Tính khoảng cách giữa vị trí xe với 4 điểm trên 4 hướng đi (ký hiệu: - - - )
- Dựa trên khoảng cách nhỏ nhất tính được, chúng tôi xác định được xe đang đi chuyển trên hướng đi nào.
- Từ đó, tăng/giảm số lượng xe tại 4 hướng tương ứng trên Firebase.
- Tổng số xe của 4 hướng, được cộng dồn vào điểm trung tâm.

B2: Nếu xe ra khỏi bán kính R, thì giảm số lượng xe tại hướng đó và của điểm trung tâm.

## 2.5. Phát triển giải thuật cho toàn hệ thống

Trong bài báo này, chúng tôi viết giải thuật xử lý đếm số lượng xe hoàn toàn trên thiết bị Android. Ứng dụng Android có nhiệm vụ: cập nhật vị trí của mình lên Firebase; tìm kiếm, thêm/bớt được các điểm giao lộ cần kiểm soát; theo dõi được số lượng xe tại các hướng của các giao lộ; bật chế độ chạy ngầm (tức người dùng có thể tắt ứng dụng, khóa máy, nhưng ứng dụng vẫn duy trì hoạt động); cảnh báo liên tục cho tài xế khi xe đi gần điểm có lượng xe đông quá ngưỡng cài đặt.

Thuật toán xử lý được xây dựng theo giải thuật sau:

B1: Khởi tạo biến, dữ liệu.

B2: Kết nối và lấy dữ liệu từ Firebase.

B3: Sử dụng API của google map để hiển thị bản đồ lên ứng dụng.

B4: Duyệt danh sách các khu vực, giao lộ cần theo dõi (Có thể tìm kiếm trên giao diện bản đồ và thêm mới)

B5: Ứng dụng Android tiến hành đọc vị trí hiện tại của mình, tính khoảng cách từ mình đến tọa độ các giao lộ (công thức (1), (2), (3)), với chu kỳ xử lý là 7s/lần.

- Nếu vị trí xe đi vào hay đi ra khỏi bán kính R của một giao lộ, thì ứng dụng xác định xe thuộc hướng đi chuyển nào (theo giải thuật ở mục 2.4); và tiến hành cập nhật lại số lượng xe tại các hướng lên Firebase (công thức (4)).
- Khi xe vào giao lộ và chuyển từ hướng này sang hướng khác, thì hệ thống vẫn cập nhật lại được số lượng xe ở các hướng (như giải thuật ở mục 2.4).
- Nếu vị trí xe vẫn ở ngoài bán kính R của các giao lộ, thì hệ thống không cập nhật lại số lượng xe tại các giao lộ.

B6: Khi người dùng bật chế độ chạy ngầm, hệ thống sẽ gửi báo động khi có giao lộ gần vị trí xe ( $R < NOTIRANGE$ ) có mật độ phương tiện đạt ngưỡng báo động THRESHOLD (Hệ thống báo động cả khi tắt ứng dụng, khóa màn hình)

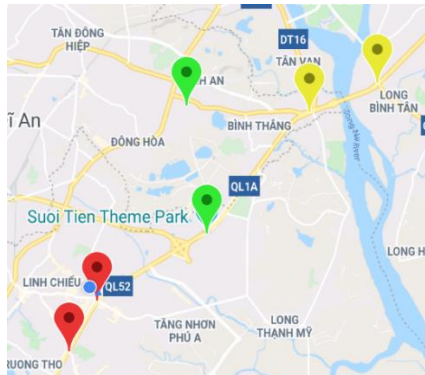
B7: Người dùng có thể mở ứng dụng và theo dõi mật độ xe trên giao diện bản đồ (Hình 4)

$$(KVnum, KVlist) = \begin{cases} KVnum ++, KVlist(add x), x \in KV \\ KVnum --, KVlist(rem x), x \in KV \end{cases} \quad (4)$$

- Trong đó: KV là khu vực cần xem xét, KVnum là số lượng xe tại các hướng của một giao lộ, KVlist là danh sách các xe kèm thời gian bị kẹt, x là tọa độ của xe đang xét.

### 3. ỨNG DỤNG ANDROID ĐÃ PHÁT TRIỂN

#### 3.1. Giao diện theo dõi các điểm giao lộ

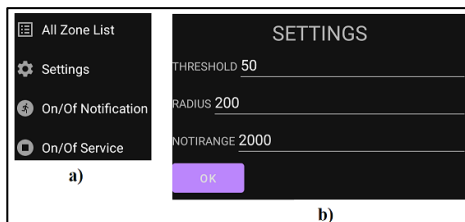


Hình 4. Giao diện ứng dụng hiển thị mật độ khu vực đang theo dõi

Hình 4 là giao diện theo dõi các giao lộ, trong đó có quy ước 3 marker màu đỏ, vàng, xanh tương ứng với mật độ báo động như sau:

-  : Mật độ xe  $\geq 8$ .
-  : Mật độ xe  $\geq 5$ .
-  : Mật độ xe  $< 5$ .

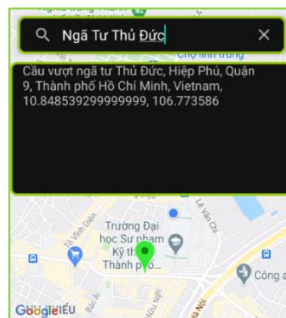
#### 3.2. Giao diện cấu hình ngưỡng



Hình 5. Giao diện cấu hình thông số

Trong Hình 5, các thông số như ngưỡng báo động (THRESHOLD), bán kính R của giao lộ (RADIUS), phạm vi theo dõi của thiết bị (NOTIRANGE) có thể cấu hình lại được.

#### 3.3. Giao diện thêm giao lộ theo dõi

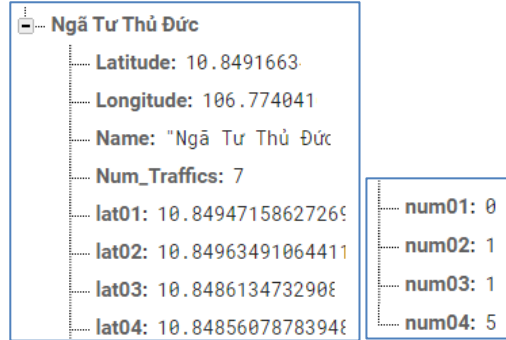


Hình 6. Giao diện tìm kiếm và thêm khu vực cần theo dõi

Ứng dụng có thể thêm các điểm giao lộ cần theo dõi như Hình 6, hoặc có thể thêm tập các giao lộ cần theo dõi bằng tay vào cơ sở dữ liệu của Firebase.

### 3.4. Bố cục dữ liệu lưu trữ trên Firebase

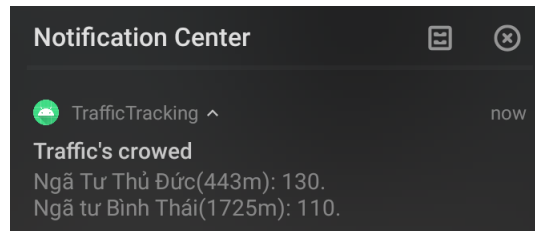
Dữ liệu được lưu trên Firebase (Hình 7).



Hình 7. Dữ liệu các khu vực giao thông được lưu trữ tại Realtime Database-Firebase

### 3.5. Chế độ báo động chạy ngầm

Khi xe di chuyển gần các điểm giao lộ bị kẹt xe (số xe cao hơn mức báo động), thì ứng dụng tiến hành gửi thông báo liên tục đến điện thoại để cảnh báo tài xế. Hệ thống báo động cả khi điện thoại khóa màn hình (Hình 8).



Hình 8. Thông báo được bật khi xe gần khu vực đông đúc (chúng tôi ngồi tại khu D, Trường Đại học Sư phạm Kỹ thuật TP.HCM)

## 4. KẾT QUẢ THỰC NGHIỆM

### 4.1. Dữ liệu thực nghiệm

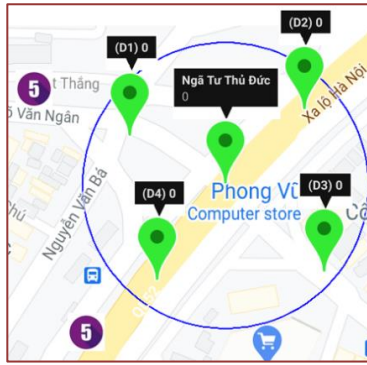
Để thực nghiệm giải thuật, cũng như ứng dụng phát triển được, chúng tôi tiến hành cài đặt ứng dụng lên 10 điện thoại Android có phần cứng cơ bản, có hệ điều hành từ 6.0 trở lên, có sóng 3G/4G, GPS của sinh viên và nhờ các bạn cầm điện thoại đi bộ ra 1 khu vực ngã tư. Trong thực nghiệm này, chúng tôi chọn ngã tư Thủ Đức. Bán kính giao lộ cầu hình là  $RADIUS(R) = 100$  m.

### 4.2. Độ đo thực nghiệm

Mọi số liệu đều được thử nghiệm trực tiếp trên 10 điện thoại thực tế.

### 4.3. Kết quả tương ứng 4 tình huống thực nghiệm

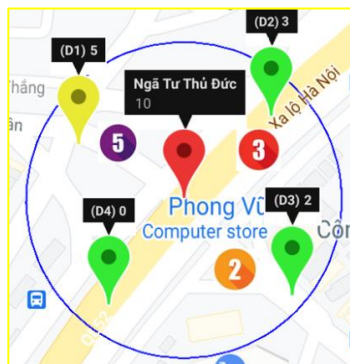
**Tình huống 1:** 10 bạn sinh viên chia thành 2 nhóm 5 bạn. Tất cả đều đứng ngoài vòng bán kính R (Hình 9)



Hình 9. Kết quả thu được trong tình huống 1

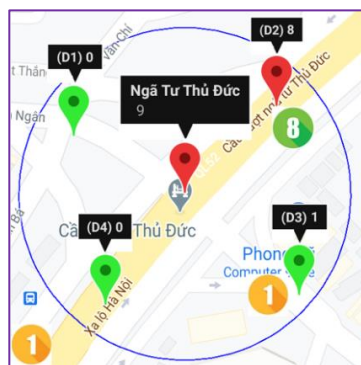
Màu sắc của các Marker được quy ước như Hình 4. Bốn Marker ở 4 hướng đi đều bằng 0, do chưa có thiết bị nào vào giao lộ. Marker ở giữa hiển thị tên giao lộ và tổng số xe hiện đang có tại giao lộ này cũng bằng 0.

**Tình huống 2:** 10 bạn cùng đi vào trong bán kính R của ngã tư Thủ Đức. Chia làm 3 nhóm ở 3 hướng đi như Hình 10.



Hình 10. Kết quả thu được trong tình huống 2

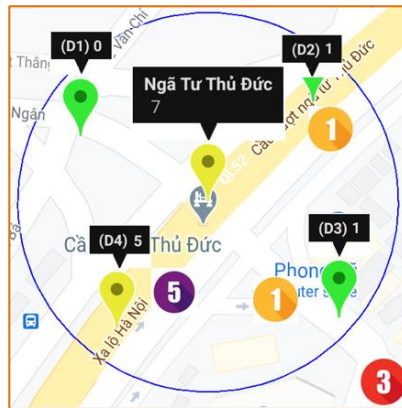
**Tình huống 3:** 9 bạn đi vào bán kính R và 1 bạn ở ngoài bán kính như Hình 11.



Hình 11. Kết quả thu được trong tình huống 3

**Tình huống 4:** 7 bạn đi vào bán kính R và 3 bạn ở ngoài bán kính như Hình 12.





Hình 12. Kết quả thu được trong tình huống 4

### Tình huống phương tiện kẹt tại tâm ngã tư:

Khi các bạn đi vào gần tâm của ngã tư, thì tùy thuộc vào khoảng cách giữa vị trí thiết bị với 4 Marker xung quanh, thì xe đó sẽ thuộc 1 trong 4 hướng này. Chúng tôi chưa phân loại thêm trường hợp ở giữa. Tuy nhiên, tổng số xe trong giao lộ (ở Marker chính giữa) vẫn hiển thị đúng số lượng xe tại giao lộ. Và các xe gần hướng nào hơn sẽ được cập nhật vào hướng đó.

### 4.4. Đánh giá giá độ chính xác

Ở các tình huống 1, 2, 3, 4, thì số lượng xe tại 4 hướng đi, và tổng số xe hiển thị ở Marker giữa đều được tự động cập nhật (cài đặt là 5 s/lần) chính xác khi các bạn sinh viên di chuyển vào và ra khu vực giao lộ. Bên cạnh đó, số lượng xe các hướng vẫn cập nhật chính xác khi các bạn di chuyển từ hướng này sang hướng khác một cách tự động. Kết quả thực nghiệm được tổng kết trong Bảng 2. Số lượng xe, màu báo động của Marker ở các hướng được hiển thị trên ứng dụng đạt độ chính xác 100%. Ứng dụng báo động chính xác 100% khi số lượng xe đạt ngưỡng cấu hình. Độ trễ hiển thị trên ứng dụng trong các tính huống có sự khác nhau, vì phụ thuộc vào tốc độ internet trên điện thoại cài đặt ứng dụng.

Bảng 2. Kết quả thực nghiệm

	Số thiết bị Android	Độ chính xác hiển thị	Độ chính xác báo động khi đông xe	Độ trễ (*)
Tình huống 1	10	100%	100%	<20s
Tình huống 2	10	100%	100%	<10s
Tình huống 3	10	100%	100%	<15s
Tình huống 4	10	100%	100%	<20s

## 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Trong nghiên cứu này, chúng tôi đã xây dựng được giải thuật xác định được số lượng xe di chuyển tại các hướng đi của một ngã tư bất kỳ. Từ đó xây dựng một ứng dụng cơ bản để chứng minh giải thuật, được cài đặt trên thiết bị Android có GPS, sóng 3G/4G. Ứng dụng có thể thêm/bớt điểm giao thông cần quan sát; cập nhật liên tục khoảng cách, số lượng xe tại các hướng của các điểm ngã tư cần theo dõi và bật báo động cho tài xế khi di chuyển trong phạm vi gần các điểm đông xe.

Song song với các chức năng mà hệ thống xử lý được thì đề tài vẫn có nhiều điểm hạn chế như: giải thuật được xử lý trực tiếp trên chính các thiết bị Android, nên nếu số lượng xe nhiều thì cần phải có máy chủ mạnh mới đáp ứng được tốc độ xử lý số lượng lớn dữ liệu; giải thuật có thể áp dụng cho tất cả các loại giao lộ: như ngã ba, ngã tư, ngã năm,.. nhưng để xử lý tốt về tất cả các trường hợp, thì thuật toán cần phải xây dựng trên một máy chủ đủ mạnh.

Với những kết quả đạt được cũng như những điểm hạn chế đã nêu, trong hướng phát triển tiếp theo của đề tài, chúng tôi sẽ xây dựng lại giải thuật trên 1 máy chủ cấu hình mạnh, nhằm xử lý được khối lượng dữ liệu lớn khi có nhiều xe tham gia giao thông; khi áp dụng vào thực tế, chúng tôi sẽ đánh giá lại lượng dữ liệu internet lấy từ sóng 3G/4G của thiết bị Android mà hệ thống sẽ mất khi so sánh với các phương pháp khác (như đặt trụ phát wifi tại các giao lộ); với các giao lộ có nhiều hướng (ngã 3, ngã 4, ngã 6, v.v.), chúng tôi sẽ xây dựng hệ thống có thể cập nhật tọa độ tại các hướng vào giao lộ và dựa trên tọa độ hiện tại của xe để xác định xe đang ở hướng nào của giao lộ; với các giao lộ có cầu vượt, chúng tôi sẽ tận dụng thêm cảm biến độ cao có sẵn trên điện thoại thông minh để xác định xe ở phía trên hay bên dưới cầu vượt; chúng tôi sẽ nghiên cứu thêm thuật toán vào việc tính toán tốc độ lưu thông của phương tiện theo các hướng. Ngoài ra, để thuận tiện cho việc tích hợp nhiều tính năng khác cho phương tiện giao thông, chúng tôi sẽ phát triển 1 khối thiết bị gắn trực tiếp trên xe với các cảm biến cần thiết để theo dõi và hỗ trợ giao thông được tốt hơn.

## TÀI LIỆU THAM KHẢO

1. Sun Ye - Research on urban road traffic congestion charging based on sustainable development, *Physics Procedia* **24** (2012) 1567-1572.
2. Vipin Jain, Ashlesh Sharma - Road traffic congestion in the developing world, *Proceedings of the 2nd ACM Symposium on Computing for Development*, 2012.
3. Wadud.Z, Zeb A., Jan S., Ayaz S., Ahmad A.M. - Best route selection for automobile traffic congestion using Android App, *Technical Journal, University of Engineering and Technology (UET) Taxila, Pakistan* Vol. **24** (4) (2019).
4. Chandana K.K, Meenakshi Sundaram S., Cyana D'sal, Meghana N. Swamy, Navya K. - A smart traffic management system for congestion control and warnings using Internet of Things (IoT), *Saudi Journal of Engineering and Technology* **2** (5) (2017) 192-196.
5. Ayesha Atta, Sagheer Abbas, Adnan Khan M., Gulzar Ahmed, Umer Farooq - An adaptive approach: Smart traffic congestion control system, *Journal of King Saud University - Computer and Information Sciences* **32** (9) (2020) 1012-1019.
6. Sandor Dornbush, Anupam Joshi - StreetSmart Traffic: Discovering and Disseminating Automobile Congestion Using VANET's, *IEEE 65th Vehicular Technology Conference*, 2007.
7. [https://www.pngitem.com/middle/imhhRow\\_android-smartphone-png-transparent-png](https://www.pngitem.com/middle/imhhRow_android-smartphone-png-transparent-png), March, 2021.
8. <https://www.javatpoint.com/firebase-realtime-database>, March, 2021.
9. <https://in.pinterest.com/pin/391461392600809918/>, March, 2021.
10. <https://www.pinterest.com/pin/839358449278826122/>, March, 2021.
11. <https://www.hararepost.co.zw/en/the-news/local-news>, March, 2021.
12. <https://www.movable-type.co.uk/scripts/latlong.html>, March, 2021.

## ABSTRACT

### DEVELOPING THE ALGORITHM FOR MONITORING AND WARNING THE VEHICLE DENSITY IN DIRECTIONS OF CROSSROAD USING ANDROID'S GPS

Huynh Hoang Ha

*Ho Chi Minh City University of Technology and Education*

Email: *hahh@hcmute.edu.vn*

Currently, traffic jam is happening quite seriously during rush hour at intersections, industrial areas, being repaired roads or due to a collision in traffic, etc. So, in this research paper, the authors develop an algorithm to track traffic density in the directions of the crossroads and warn the driver when there is a traffic jam ahead. The authors use GPS system (global positioning system) on the Android devices. These devices are attached on the traffic vehicle or carried with the driver (Android phone), thereby calculating and synchronizing the number of traffics of the crossroad to Firebase. The application is installed the algorithm which can add and remove the intersections to be tracked, identify the number of vehicles in the directions of the crossroad, calculate the distance between the vehicle and the crossroads and send warning to the driver when there is a crowded spot. Vehicle density is accurately determined without depending on environmental factors such as dark, rainy, etc.

*Keywords:* GPS, Android, firebase, traffic jam, traffic.