

## MỘT GIẢI PHÁP CẢI TIẾN CƠ CHẾ ĐỊNH TUYẾN DSR DỰA TRÊN TÁC TỬ DI ĐỘNG TRONG MẠNG MANET

CUNG TRỌNG CƯỜNG<sup>1</sup>, VÕ THANH TÚ<sup>2</sup>, NGUYỄN THỨC HẢI<sup>3</sup>

<sup>1</sup>Trường Cao đẳng Công nghiệp Huế; *ctcuong@hueic.edu.vn*

<sup>2</sup>Trường Đại học Khoa học, Đại học Huế

<sup>3</sup>Trường Đại học Bách khoa Hà Nội

**Tóm tắt.** Bài báo phân tích về hoạt động của các cơ chế định tuyến AODV, DSR trong mạng tùy biến không dây (MANET). Từ đó, đề xuất một cơ chế định tuyến mới MAR-DSR dựa trên tác tử di động để nâng cao hiệu năng mạng trong môi trường có mật độ lớn và độ di động cao. Tập trung chính vào việc cải tiến cơ chế cập nhật trạng thái thích nghi và khả năng phán đoán đường đi của mỗi nút. Cơ chế định tuyến sử dụng tác tử được thực hiện trong bài báo là MAR-DSR, được cài đặt trên OMNeT++ cho kết quả đánh giá hiệu năng so với các giải thuật chuẩn DSR.

**Từ khóa.** Hệ thống tác tử di động, MANET, thuật toán tối ưu, mạng tùy biến không dây.

**Abstract.** In this article, we focus on studying basic features of Mobile Agent system to improve routing mechanism in Mobile Ad hoc Network (MANET). Based on mobile agent, the MAR-DSR model and algorithm are proposed to optimize network capacity in highly mobile environment. The best updating algorithm for routing are based on a congestion analysis unit and route anticipating capability of each network node. Simulation on software is used to assess effectiveness of algorithm compared to DSR.

**Key words.** Mobile agent system, MANET, optimal routing, Ad hoc networks.

### 1. GIỚI THIỆU

Vấn đề truyền thông tin trong mạng không dây đóng vai trò quan trọng trong hầu hết các lĩnh vực, đặc biệt với sự phát triển của các dịch vụ truyền thông đa phương tiện đã làm cho lưu thông trên đường truyền càng lớn và phổ biến. Đối với một số ứng dụng đòi hỏi tính di động cao và mật độ truyền lớn thì khả năng đáp ứng của cơ chế định tuyến thích nghi như AODV, DSR [5], vẫn còn một số hạn chế. Vì vậy, trong những năm gần đây, các nhà nghiên cứu đã cố gắng nâng cao tính sẵn sàng và tin cậy trong bài toán định tuyến thích nghi để đáp ứng nhanh với sự di động của hệ thống [8]. Một trong những giải pháp được sử dụng trong bài báo này là tác tử di động (mobile agent), sử dụng đặc tính tự trị và khả năng di động từ nút này sang nút khác để hoàn tất tác vụ [6]. Ý tưởng chính của tác tử di động là di chuyển xử lý đến gần nguồn dữ liệu, nhờ đó có thể giảm tải mạng, khắc phục tình trạng trễ, hỗ trợ xử lý không đồng bộ và tạo ra sự tương thích mạnh trên các môi trường không đồng nhất

[2]. Tác tử di động với các ưu điểm này hứa hẹn một giải pháp mới, hiệu quả và dễ dàng hơn trong việc phát triển ứng dụng phân tán.

Đặc trưng cơ bản nhất của các mạng di động là mỗi nút mạng đều có khả năng di chuyển. Vì vậy, vấn đề cập nhật thông tin trạng thái mạng tại mỗi nút và mỗi nhóm di động để có cơ chế truyền, nhận và định tuyến dữ liệu một cách tối ưu là điều đặc biệt quan trọng. Với phương thức định tuyến điều khiển theo yêu cầu, khi có một yêu cầu từ nguồn đến đích, nút nguồn phải khởi đầu một quá trình định tuyến, quá trình này chỉ hoàn tất khi đã tìm ra một lộ trình sẵn sàng hoặc tất cả các lộ trình khả thi đều đã được kiểm tra [13]. Khi một lộ trình đã được tìm ra và thiết lập, nó được duy trì bởi một số dạng thủ tục cho đến khi hoặc là lộ trình đó không thể truy nhập được từ nút nguồn hoặc là lộ trình đó không cần thiết nữa. Do vậy, việc cài đặt các tác tử di động và thông minh là cần thiết để cải thiện chức năng định tuyến trên mạng MANET [10,15].

## 2. GIAO THỨC ĐỊNH TUYẾN THEO YÊU CẦU TRONG MẠNG MANET

Như đã biết, việc định tuyến trên các hệ thống mạng là khá quan trọng, quá trình định tuyến có thể xảy ra trước khi hệ thống có nhu cầu truyền dữ liệu hoặc trong khi hệ thống truyền dữ liệu. Định tuyến điều khiển theo yêu cầu là một trong những phương thức định tuyến chỉ xảy ra khi hệ thống có nhu cầu truyền dữ liệu. Định tuyến theo yêu cầu được đánh giá phù hợp và có ưu điểm trong các mạng MANET, trong đó nổi bật là giao thức DSR, AODV và TORA, giao thức được phân tích, đánh giá ở đây là giao thức định tuyến DSR [3,13].

### 2.1. Giao thức DSR (Dynamic Source Routing)

Giao thức DSR là giao thức định tuyến đơn giản và hiệu quả được thiết kế riêng cho mạng MANET. Nó sử dụng cơ chế định tuyến nguồn (source routing), cho phép mạng tự động tổ chức và cấu hình mà không cần đến sự can thiệp của người quản trị hoặc cơ sở hạ tầng sẵn có của mạng. Phần Header của gói dữ liệu sẽ lưu trữ thứ tự các nút mà cần phải đi qua để đạt tới đích. Do vậy, các nút trung gian chỉ cần giữ liên lạc với các nút láng giềng của nó để chuyển tiếp các gói tin. Tại mỗi một nút trong mạng luôn duy trì một bộ nhớ đệm (Router Cache), các gói tin sẽ nhận thông tin về đường đi và thực hiện việc truyền tin trên đường đi đã chọn. Ngược lại, khi không tồn tại đường đi trong Router Cache hoặc có tồn tại đường đi nhưng không còn hiệu lực, DSR sẽ thực hiện cơ chế phát hiện đường (Route Discovery) bằng cách gửi các gói tin quảng bá Route Request đến các nút lân cận trên toàn bộ mạng. Khi đường đi được tìm thấy, gói tin Route Reply sẽ chứa thứ tự các chặng tới đích và được truyền trở lại nguồn [14].

Như vậy, hoạt động của giao thức DSR bao gồm hai cơ chế chính: cơ chế tạo thông tin định tuyến (Route Discovery) và cơ chế duy trì thông tin định tuyến (Route Maintenance) với thuật toán cơ chế xử lý khám phá đường đi tại nút của DSR:

*Bước 1:* Thông qua trường request ID, nó sẽ kiểm tra xem đã nhận gói tin này hay chưa? Nếu đã tồn tại thì nó sẽ loại bỏ gói tin đó và không xử lý gì thêm. Ngược lại thì qua bước 2.

*Bước 2:* Kiểm tra trong Route Cache của nó có đường đi đến node đích mà còn hiệu lực hay

không? Nếu có đường đi đến đích thì nó sẽ phản hồi lại cho node nguồn bằng gói Route Reply (RREP) chứa thông tin về đường đi đến đích và kết thúc tiến trình. Ngược lại thì qua bước 3.

*Bước 3:* Kiểm tra địa chỉ đích cần tìm có trùng với địa chỉ của nó hay không? Nếu trùng thì nó gửi lại cho node nguồn gói Route Reply (RREP) chứa thông tin về đường đi đến đích và kết thúc tiến trình. Ngược lại thì nó sẽ phát broadcast gói tin RREQ đến các node láng giềng của nó. Các nút láng giềng sau khi nhận gói tin RREQ sẽ thực hiện việc kiểm tra thông tin (quay về bước 1).

Như vậy, quá trình này cứ tiếp tục cho đến khi nút nguồn nhận được thông tin về đường đi đến đích hoặc thông tin rằng không thể định tuyến đến đích. Gói Route Reply (RREP) được gửi đến nguồn bằng cơ chế phát Unicast với Source Route là đảo ngược Source Route trong gói RREQ.

## 2.2. Đánh giá một số nhược điểm của giao thức DSR

Từ việc phân tích cơ chế hoạt động của DSR, một số nhận xét về thuật toán như sau: Tiến trình khám phá đường đi được thực hiện dựa trên việc gửi quảng bá và nhận phản hồi. Thông tin định tuyến được lưu trữ tại tất cả các nút trung gian. Trong quá trình khám phá tuyến đường, các nút trung gian đều có khả năng học đường về đích hoặc ngược về nguồn.

Ngoài ra, giao thức DSR còn có một số nhược điểm: tại mỗi nút luôn duy trì thông tin về toàn bộ đường đi về đích hoặc về nguồn, do đó khi có vấn đề nảy sinh về lỗi của đường đi hoặc vấn đề tắc nghẽn cục bộ tại một điểm nút nào đó trong đường đi đã xác định thì sẽ xảy ra vấn đề rơi gói tin hoặc lỗi truyền không xác định trước, vấn đề tương tự đối với bảng thông tin cập nhật trong lộ trình đường đi [13].

DSR sử dụng cơ chế định tuyến nguồn, theo đó nó luôn trả lời cho tất cả các yêu cầu tìm đường. Cơ chế này giúp DSR thu thập được nhiều đường đi về đích dẫn đến khả năng phát tin tốt hơn AODV. Tuy nhiên, điều này chỉ tốt trong trường hợp mạng có ít nguồn phát và mức độ di chuyển không cao, trong trường hợp mức độ di chuyển cao khả năng các nút sẽ bị mất liên lạc với nhau nhiều là nguyên nhân dẫn đến số lượng đường đi mất hiệu lực trong bộ nhớ đệm tăng thêm vào đó là sự gia tăng các thông điệp Reply dẫn đến giảm sút hiệu suất của DSR.

Như vậy, thuật toán này khi duy trì lộ trình không quan tâm đến trạng thái của các nút trên lộ trình. Cụ thể là, khi có yêu cầu đến, nếu đã có lộ trình trong bộ nhớ đệm thì tiến hành truyền ngay, cho dù có tồn tại một nút trung gian nào đó trên lộ trình đã bị nghẽn hoặc đã mất liên kết (nhưng chưa cập nhật lại lộ trình). Khi truyền đến nút này thì sẽ xảy ra tình trạng tắc nghẽn và đây là nhược điểm của thuật toán cơ bản của các thuật toán này cần phải cải tiến.

## 2.3. Một số nghiên cứu liên quan

Để cải tiến các thuật toán định tuyến DSR cũng như một số thuật toán định tuyến AODV, TORA..., một số nghiên cứu trên thế giới đã đề xuất các giải pháp cải tiến trong đó giải pháp cải tiến giao thức DSR [1,11] theo các tham số về đo mức năng lượng của mỗi nút (mức tín hiệu) và metric của mỗi nút, hoặc cải tiến giao thức các giao thức AODV bằng thuật toán cải tiến khả năng di động [5], tuy nhiên việc cải tiến chỉ giải quyết một số trường hợp cụ thể và

chưa giải quyết hiệu quả trường hợp mật độ truyền dữ liệu cao và các nút di động hoặc thực hiện cải tiến trên toàn bộ trường hợp. Như vậy, các nghiên cứu về cải tiến các thuật toán cơ bản của giao thức định tuyến theo yêu cầu trong các trường hợp, môi trường khác nhau đang là vấn đề được nhiều nhóm nghiên cứu quan tâm và nghiên cứu, hiện tại một số kết quả nghiên cứu trên thế giới cũng đã công bố và nhiều nghiên cứu đang thực hiện trên nhiều hướng khác nhau để cải thiện cho những mô hình, trường hợp khác nhau đối với mạng MANET.

### 3. ĐỀ XUẤT GIẢI THUẬT ĐỊNH TUYẾN CHO MẠNG MANET DỰA TRÊN CÔNG NGHỆ TÁC TỬ DI ĐỘNG

#### 3.1. Ý tưởng của giải thuật

Qua phân tích ở trên, trong khuôn khổ của bài báo này sẽ tập trung nghiên cứu tích hợp tác tử di động vào thuật toán định tuyến cơ bản trong mạng MANET là DSR để nâng cao hiệu quả sử dụng tài nguyên mạng. Thuật toán cải tiến là MAR-DSR (Mobile Agent Routing - DSR), trong đó tác tử di động thực hiện hai chức năng cơ bản sau đây:

- Xác định trạng thái của mỗi nút mạng để cập nhật thông tin cho giai đoạn khám phá lộ trình trong thuật toán DSR. Trạng thái của nút mạng được xác định qua nhiều tham số, như xác suất tắc nghẽn, lưu lượng phát sinh tại nút đó, chiều dài bộ đệm,... Trong bài báo này, trạng thái nút mạng được xác định bởi tham số xác suất nghẽn tại mỗi nút.
- Dựa trên tham số về trạng thái nút mạng thông qua tác tử di động, thuật toán MAR-DSR và sẽ quyết định lựa chọn lộ trình trong Route cache hay thực hiện khám phá lại lộ trình.

#### 3.2. Mô tả giải thuật

Giải thuật MAR-DSR cũng được thực hiện qua 2 giai đoạn là khám phá lộ trình và duy trì lộ trình như các giải thuật DSR. Tuy nhiên, mỗi giai đoạn đều được điều khiển bởi tác tử di động chứa thông tin trạng thái mỗi nút mạng để tối ưu hóa việc định tuyến.

##### 3.2.1. Khám phá lộ trình

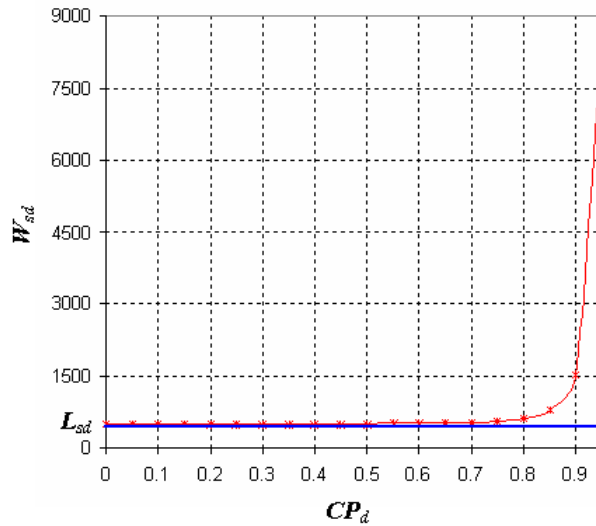
- Khi nút nguồn muốn gửi dữ liệu đến một đích nào đó, đầu tiên nó phải xem trong bộ nhớ đệm (cache) đã có lộ trình cần tìm hay chưa. Nếu không tìm thấy đường đi đến nút đích, nó bắt đầu hoạt động khám phá lộ trình. Nếu lộ trình đã tìm thấy trong bộ nhớ cache nhưng mức độ tắc nghẽn tại các nút trung gian vượt quá ngưỡng cho phép thì cũng phải thực hiện khám phá lại lộ trình.
- Giai đoạn đầu tiên của khám phá định tuyến được bắt đầu bằng cách gửi một số gói tin đến các nút liền kề, công việc này thực hiện bằng việc phát quảng bá các tác tử (agent). Các agent này được gọi là Forward Agent (FA).
- Cơ chế lựa chọn lộ trình: Để lựa chọn lộ trình tối ưu, ta thiết lập một hàm trọng số cho các kết nối dựa trên 2 tham số cơ bản: xác suất tắc nghẽn tại mỗi nút và khoảng cách giới hạn về công suất thu. Hàm trọng số được thiết lập như sau:

$$w_{sd} = L_{sd} + \frac{1}{(1 - CP_d)^3}, \quad (3.1)$$

trong đó,  $L_{sd}$  là khoảng cách giới hạn về công suất thu, nghĩa là khoảng cách giới hạn để nút nguồn  $s$  nhận được năng lượng của nút đích  $d$ .  $CP_d$  (Congestion Probability) là xác suất nghẽn tại nút  $d$ . Các thuật toán MAR-DSR sẽ lựa chọn lộ trình tối ưu sao cho trọng số nhỏ nhất.

Ta thấy rằng, với hàm trọng số được thiết lập như hàm (3.1), khi mức độ tắc nghẽn của nút lớn thì trọng số của kết nối tăng, do vậy xác suất lựa chọn lộ trình đi qua nút đó giảm, sẽ lựa chọn lộ trình phù hợp hơn với mức độ tắc nghẽn nhỏ hơn.

Bài báo đề xuất hàm (3.1) với mục tiêu thay đổi trọng số  $w_{sd}$  từ nút  $s$  đến nút  $d$  theo khoảng cách  $L_{sd}$  và mức độ tắc nghẽn  $CP_d$  của nút  $d$ , trong đó,  $CP_d$  được tính toán bởi tác tử di động. Với hàm trọng số (3.1), ta có sự thay đổi của  $w_{sd}$  theo  $L_{sd}$  và  $CP_d$  như ở hình 1.



Hình 1. Sự thay đổi của Wsd theo mức độ tắc nghẽn của nút

Từ kết quả trên hình 1 ta thấy rằng, khi mức độ tắc nghẽn của nút ( $CP_d$ ) nhỏ hơn 75% thì trọng số giữa các nút phụ thuộc chủ yếu vào khoảng cách giữa các nút đó. Còn khi  $CP_d$  lớn hơn 80% thì trọng số sẽ tăng lên rất lớn, do vậy nút sẽ bị loại bỏ ra khỏi lộ trình tuyến dữ liệu. Vì vậy, trong mô hình đề xuất, ngưỡng  $CP_d$  được thiết lập là 0.75.

$CP_d$  được tính dựa trên số liệu thống kê trong quá trình mô phỏng, bằng tổng số gói nghẽn tại node  $d$  chia cho tổng số gói truyền đến node  $d$  tại thời điểm đang xét. Các giá trị này được cập nhật thường xuyên bởi các tác tử di động.

*Mô tả các bước của thuật toán khám phá lộ trình như sau:*

*Bước 1:* Phân tích trường request ID, kiểm tra xem đã nhận gói tin này hay chưa? Nếu đã tồn tại thì nó sẽ loại bỏ gói tin đó và dừng. Ngược lại thì qua bước 2;

*Bước 2:* Kiểm tra trong Route Cache của nó có đường đi đến node đích mà còn hiệu lực hay không? Nếu có đường đi đến đích thì nó sẽ phản hồi lại cho node nguồn bằng gói Route Reply

(RREP - bằng cách thực hiện bởi gói FA kèm theo) chứa thông tin về đường đi đến đích và kết thúc tiến trình. Ngược lại thì qua bước 3;

*Bước 3:* Kiểm tra giá trị CP tại vị trí láng giềng kế tiếp; tính toán giá trị CP tại nút kế tiếp theo các đường đi được cập nhật; Nếu gặp tình trạng không kết nối; trở lại bước 1; ngược lại thì qua bước 4;

*Bước 4:* Kiểm tra địa chỉ đích cần tìm có trùng với địa chỉ của nó hay không? Nếu trùng thì nó gửi lại cho node nguồn gói Route Reply (RREP - bằng cách thực hiện bởi gói FA kèm theo) chứa thông tin về đường đi (vào Router Cache) đến đích và kết thúc tiến trình. Ngược lại thì nó sẽ phát broadcast gói tin RREQ đến các node láng giềng của nó. Các nút láng giềng sau khi nhận gói tin RREQ sẽ thực hiện việc kiểm tra thông tin. Xác định giá trị của CP tại nút xem xét, cập nhật CP mới và lưu trong bộ đệm Route cache.

Như vậy, việc chọn lựa lộ trình dựa vào xác suất tắc nghẽn và khoảng cách giới hạn (bước 2) thông qua giá trị kết quả của  $w_{sd}$ .

Trong thuật toán này, các gói tin RREP và RREQ được giữ nguyên như trong thuật toán DSR. Cấu trúc của các gói tin FA và BA được mô tả như sau:

Cấu trúc tác tử FA gồm có 3 trường như ở bảng 1, chức năng của các trường được mô tả như sau:

- ID: Số thứ tự của yêu cầu khám phá lộ trình.
- Src\_ID: Địa chỉ nút nguồn của lộ trình cần khám phá.
- Dest\_ID: Địa chỉ nút đích của lộ trình cần khám phá.

*Bảng 1.* Cấu trúc gói tin FA

ID	Src_ID	Dest_ID
16 bits	8 bits	8 bits

Cấu trúc tác tử BA gồm có 3 trường như ở bảng 2, chức năng của các trường được mô tả như sau:

- ID: Số thứ tự của yêu cầu khám phá lộ trình.
- Intermediate\_ID: Địa chỉ của các nút trung gian trên lộ trình cần khám phá.
- CP: mức độ tắc nghẽn của nút trung gian đang xét.

*Bảng 2.* Cấu trúc gói tin BA

ID	Src_ID	Intermediate_ID	CP
16 bits	8 bits	8 bits	double

Xét một ví dụ như ở hình 2. Giả sử có một yêu cầu khám phá lộ trình để truyền dữ liệu từ nút 1 đến nút 6, và đây là yêu cầu thứ 50. Tình trạng tắc nghẽn của các nút ở thời điểm hiện tại là các giá trị được ghi tại các nút mạng. Cơ chế cập nhật thông tin trạng thái của các nút mạng bằng tác tử di động được thực hiện như sau:

- Đầu tiên, nút nguồn (nút 1) gửi các agent FA đến các nút láng giềng của nó với cấu trúc FA(50, 1, 6), nghĩa là khám phá lộ trình cho yêu cầu thứ 50 từ nút 1 đến nút 6.
- Các nút láng giềng sau khi nhận được agent FA sẽ phân tích thông tin để xem có phải là nút đích hay không, nếu không, tiếp tục gửi FA đến các nút láng giềng tiếp theo. Lặp

lại cho đến khi gửi đến nút đích (hình 2a).

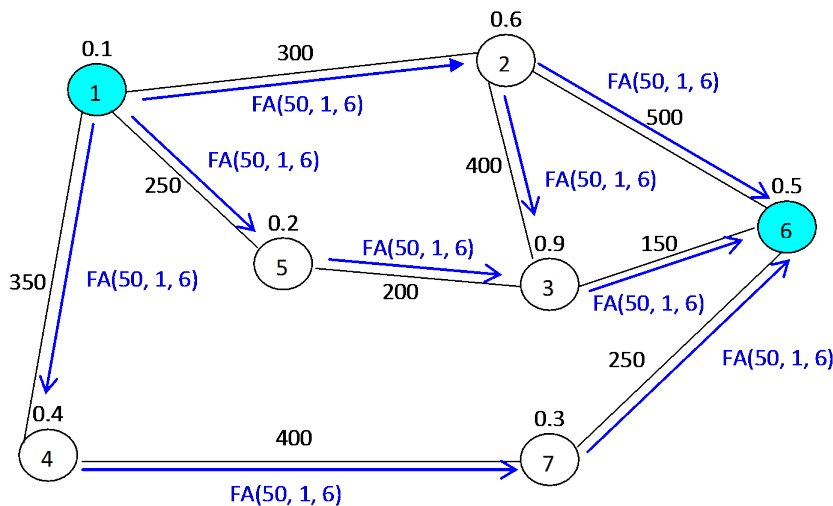
- Sau khi tác tử FA được gửi đến đích, nút đích sẽ thực hiện gửi về nút nguồn các tác tử BA theo các lộ trình đã chọn để cập nhật thông tin về tình trạng tắc nghẽn tại mỗi nút tại thời điểm hiện tại (hình 2b).
- Dựa trên thông tin của tác tử BA, thuật toán MAR-DSR sẽ thực hiện tính toán lại trọng số của các kết nối theo phương trình 3.1, sau đó lựa chọn lộ trình có trọng số nhỏ nhất.

Trong trường hợp như ở hình 2, trọng số của các kết nối sẽ được cập nhật lại sau khi nút nguồn nhận được tác tử BA như sau:

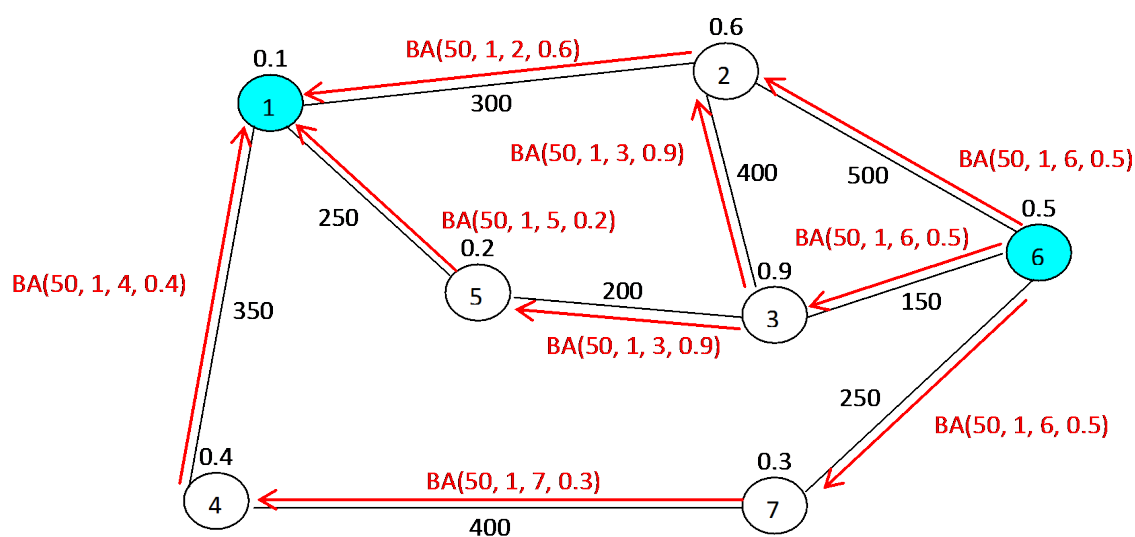
Lộ trình	Trọng số các hop theo hàm 3.1			Tổng trọng số
1 → 2 → 6	1 → 2	2 → 6		823.6
	310.6	508		
1 → 2 → 3 → 6	1 → 2	2 → 3	3 → 6	1873.6
	310.6	1400	158	
1 → 5 → 3 → 6	1 → 5	5 → 3	3 → 6	1610
	252	1200	158	
1 → 4 → 7 → 6	1 → 4	4 → 7	7 → 6	1015.5
	354.6	403	258	

Từ kết quả trên, lộ trình được chọn là 1 → 2 → 6.

Ta thấy rằng, nếu không sử dụng tác tử di động để cập nhật thông tin về trạng thái tắc nghẽn tại các nút thì lộ trình được chọn sẽ là 1 → 5 → 3 → 6 (vì khoảng cách nhỏ nhất). Nếu chọn lộ trình này thì khả năng nghẽn tại nút 3 là rất lớn. Với thuật toán MAR-DSR, ta chọn lộ trình tối ưu là 1 → 2 → 6 sẽ hạn chế được xác suất nghẽn mạng.



Hình 2a



Hình 2b

Hình 2. Minh họa nguyên lý hoạt động của thuật toán MAR-DSR

### 3.2.2. Duy trì đường đi

Duy trì đường đi là vai trò rất quan trọng của MANET để giữ tính thay đổi động và tìm đường đi tốt nhất trong quá trình khám phá có thể thực hiện việc điều khiển việc nghẽn mạch, độ mạnh tín hiệu, năng lượng,...

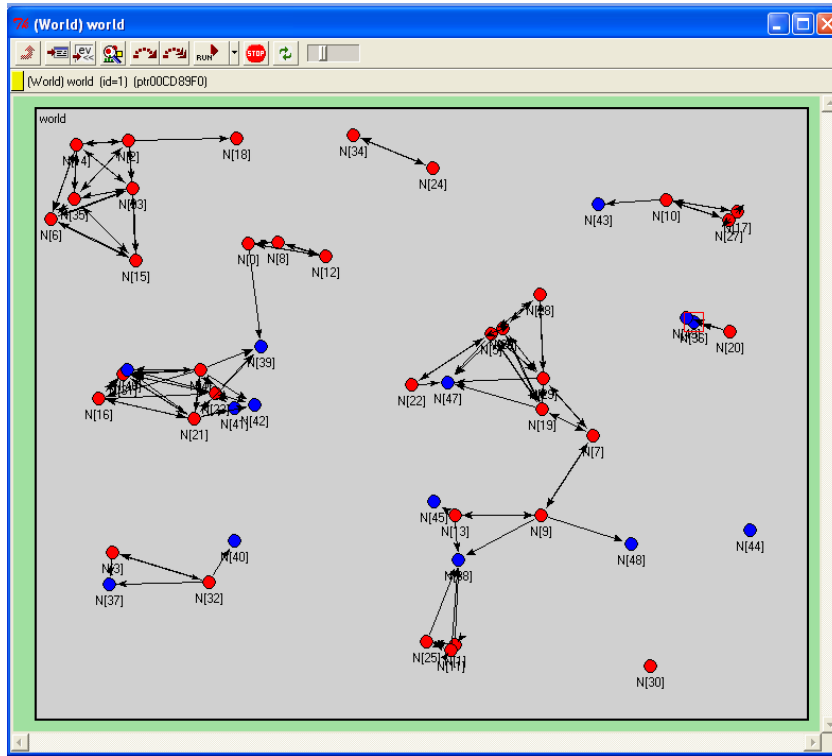
Trong các thuật toán MAR-DSR ở trên, cơ chế duy trì đường đi được thực hiện dựa trên thông tin xác suất tắc nghẽn (CP) của mỗi nút. Nếu CP vượt quá giới hạn cho phép thì thực hiện lại quá trình khám phá lộ trình. Trong bài báo này, theo thực nghiệm việc thiết lập các giá trị ngưỡng phù hợp cho xác suất tắc nghẽn, đã đánh giá và thử nghiệm giá trị trong khoảng 0.5 – 0.75, trong đó đã lựa chọn ngưỡng tối đa cho xác suất tắc nghẽn và chọn giá trị thiết lập ngưỡng giới hạn của mỗi nút là  $CP = 0.75$ .

## 4. MÔ PHỎNG VÀ ĐÁNH GIÁ KẾT QUẢ

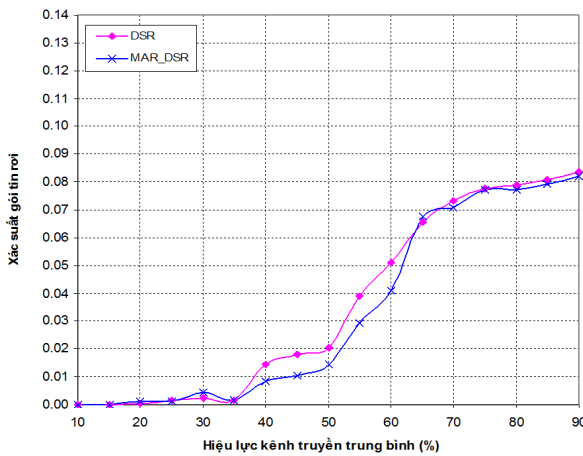
Để đánh giá hiệu quả thực thi của thuật toán MAR-DSR; ta đã cài đặt mô phỏng trên OMNeT++, so sánh với các thuật toán truyền thống là DSR đã được cài đặt trong module adHocSim. Giao diện chính của chương trình mô phỏng như hình 2, được mô phỏng trên tập dữ liệu có 50 nút. Các đường kết nối trên hình biểu thị rằng tại thời điểm đang xét chúng là các nút láng giềng của nhau.

Trong hình 5, so sánh kết quả mô phỏng thuật toán MAR-DSR với thuật toán DSR, ta thấy rằng, khi lưu lượng trung bình trên toàn mạng nhỏ, thuật toán MAR-DSR chưa có hiệu quả. Nguyên nhân là do trong trường hợp này xác suất nghẽn (CP) tại các nút nhỏ, nên trọng số của các kết nối như đã thiết lập ở phương trình (3.1) không thay đổi nhiều. Khi lưu lượng mạng trong khoảng từ 35% đến 70% thì thuật toán MAR-DSR thực thi hiệu quả hơn thuật

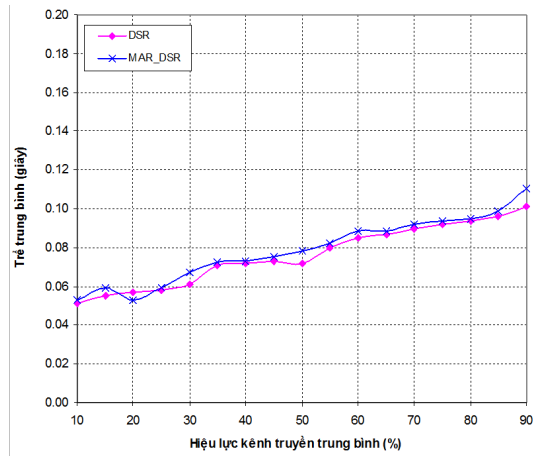




Hình 3. Giao diện chính của chương trình mô phỏng



Hình 5. Xác suất nghẽn của thuật toán MAR-DSR so với DSR

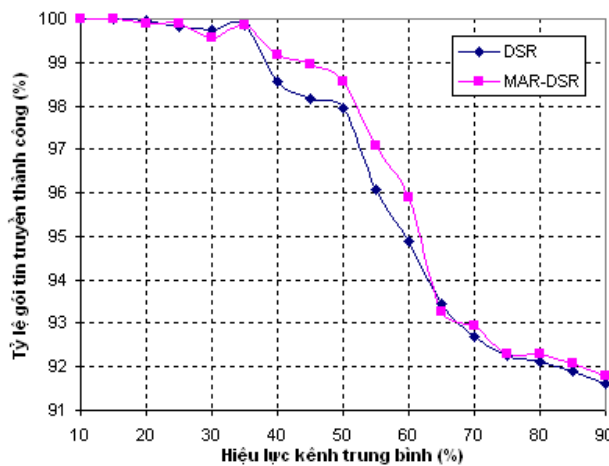


Hình 6. Trễ truyền tải gói tin trung bình

toán DSR về mặt xác suất nghẽn mạng. Trong trường hợp lưu lượng mạng lớn (>75%) thì xác suất nghẽn mạng bắt đầu tăng lên do đó hiệu quả của thuật toán MAR-DSR cũng giảm đi tuy nhiên vẫn có hiệu quả so với DSR nếu xét trên tổng gói tin thành công như kết quả được mô tả ở bảng 1.

Kết quả mô phỏng trên hình 6 là độ trễ truyền tải gói tin của thuật toán MAR-DSR so

với DSR. Ta thấy rằng, khi tích hợp tác tử di động vào điều khiển định tuyến thì làm tăng thì làm tăng độ trễ truyền tải do thời gian xử lý tại mỗi nút tăng. Tuy nhiên, độ trễ tăng lên không đáng kể, ta có thể cải tiến để tiệm cận độ trễ tương đương giao thức chuẩn.



(a)

Lưu lượng trung bình (%)	Tỷ lệ gói tin truyền thành công (%)	
	DSR	MAR-DSR
10	100.0000000	100.0000000
15	100.0000000	100.0000000
20	99.97492748	99.88049275
25	99.82099240	99.87809924
30	99.76077149	99.56077149
35	99.84031070	99.84031070
40	98.54446011	99.15444601
45	98.17345771	98.97345771
50	97.95481013	98.55481013
55	96.08087997	97.08087997
60	94.89549986	95.89549986
65	93.45425202	93.25425202
70	92.67874296	92.92787430
75	92.24787430	92.27874296
80	92.10878743	92.27874296
85	91.90878743	92.07874296
90	91.62208787	91.77874296

(b)

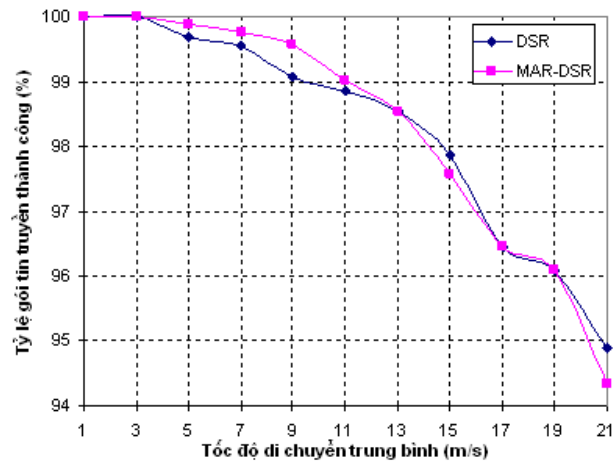
Hình 6. Tỷ lệ gói tin truyền thành công theo sự thay đổi của lưu lượng trung bình trên mạng (a) và tỉ lệ gói tin thành công trung bình trên mạng (b)

Trên hình 6, đánh giá hiệu quả thực thi của thuật toán MAR-DSR bằng tham số tỷ lệ gói tin truyền thành công. Ta thấy rằng, khi lưu lượng trung bình trên mạng từ khoảng 35% đến 75% thì thuật toán MAR-DSR cho ta tỷ lệ gói tin truyền thành công cao hơn (hình 6a), giá trị này cho ta thấy chi tiết hơn theo tỷ lệ gói tin thành công được mô tả trong bảng hình 6b.

Để đánh giá về hiệu quả khi di chuyển các nút mạng đối với thuật toán DSR, khi ta thay đổi tốc độ di chuyển trung bình của các nút mạng (hình 7) thì thuật toán MAR-DSR cũng chỉ có hiệu quả khi tốc độ di chuyển ở mức vừa phải, cụ thể là nhỏ hơn 13m/s trong trường hợp này. Còn với tốc độ di chuyển cao thì MAR-DSR chưa có cải thiện hơn thuật toán gốc. Theo kết quả mô phỏng các kết quả của thuật toán MAR-DSR với DSR, các kết quả trung bình đều tốt hơn thuật toán gốc và có hiệu quả khi mật độ mạng trung bình và cao (dưới 75%), điều này thể hiện trong các kết quả mô phỏng của thuật toán cải tiến.

## 5. KẾT LUẬN

Bài báo đã nghiên cứu các giải pháp sử dụng tác tử di động vào điều khiển các giao thức định tuyến trong mạng MANET nhằm nâng cao hiệu quả cơ chế định tuyến theo yêu cầu.



Hình 7. Đánh giá hiện quả của thuật toán khi thay đổi tốc độ di chuyển trung bình của nút

Đề xuất các thuật toán định tuyến MAR-DSR trên cơ sở cải tiến thuật toán định tuyến cơ bản nhất của mạng MANET là DSR, sử dụng tác tử di động để cải tiến cơ chế điều khiển với chức năng chính là cập nhật thông tin trạng thái mạng dựa trên thông số tắc nghẽn và khoảng cách giới hạn truyền tải. Kết quả mô phỏng cho thấy rằng việc tích hợp tác tử di động vào thuật toán DSR đã mang lại hiệu quả về mặt xác suất nghẽn mạng trong trường hợp lưu lượng trung bình trên toàn mạng ở mức vừa phải đến cao.

Trong những nghiên cứu tiếp theo, ta sẽ cải tiến cơ chế điều khiển của tác tử di động trong đó xác định hàm trọng số phù hợp hơn để cải tiến giao thức DSR, đánh giá và bổ sung các tham số khác để nâng cao hiệu quả thực thi của thuật toán, và đánh giá thuật toán cải tiến MAR-DSR và các thuật toán cải tiến cho các thuật toán định tuyến theo yêu cầu để xác định được thuật toán tối ưu đồng thời tiếp tục tích hợp vào các thuật toán định tuyến cơ chế di động thông minh trong việc xác định giá trị tắc nghẽn (chọn tham số tắc nghẽn thông minh) và tăng tính di động tại các nút của mạng MANET để tăng hiệu quả thực thi các giao thức nghiên cứu đặt ra trong bài báo.

## TÀI LIỆU THAM KHẢO

- [1] M. Rajabzadeh, F. Adibniya, M. Ghasemzadeh, Adaptive DSR protocol with cooperative agents for different mobility and traffic patterns, *The 3th International Conference on Systems and Networks Communications, ICSNC '08*, 2008 (310–315). NƠI TỔ CHỨC?
- [2] Zhipeng Liu, Chan-Heum Park, Bokman Lee, Chonggun Kim, *A Routing Agent Using Cross-Layer Method for Collision Avoidance in Ad Hoc Networks, Agent and Multi-Agent Systems: Technologies and Applications*, Springer Berlin Heidelberg, 2009 (325–334).
- [3] C.V. Mahapurush, S.S. Manvi, M.S. Kakkasageri, Performance Analysis of AODV, DSR, and Swarm Intelligence Routing Protocols In Vehicular Ad hoc Network Environment, *International Conference on Future Computer and Communication*, 2009 (21–25).
- [4] Elizabeth M. Royer et al., A review of current routing protocols for Ad hoc mobile wireless networks, *IEEE Personal Communications*, 107-9916/99, (1999) 46–55.

- [5] M. Jdrees, M. M. Yousaf, S. W. Jaffry, M. A. Pasha, . S. A. Hussain, Enhancements in AODV routing using mobility aware agents, emerging technologies, *Proceedings of the IEEE Symposium*, Punjab University College of Information Technology, 2005 (98-102).
- [6] Joseph P. Macker, William Chao, Ranjam Abramson, “Multi-Agent Systems in Mobile Ad hoc Networks”, Naval Research Laboratory, 2007.
- [7] Krzysztof Szczypiorski, Igor Margasinski, Wojciech Mazurczyk, Steganographic routing in multi agent system environment, *Journal of Information Assurance and Security* **2** (2007) 235–243.
- [8] Geetha Jayakumar, and G. Gopinath, Ad hoc mobile wireless networks routing protocols - a review, *Journal of Computer Science* **3** (8) (2007) 574–582.
- [9] Padmini Misra, Routing protocols for Ad Hoc mobile wireless networks, <http://www.cis.ohio-state.edu/~misra>.
- [10] Manal Abdullah, Helen Bakhsh, Agent-based dynamic routing system for MANETs, *ICGST-CNIR Journal* **9** (1) (2009) 27–38.
- [11] M. Rajabzadeh, F. Adibniya, M.ghasemzadeh, MA-DSR: Multi agent based adaptive dsr protocol with intelligent behavior in realistic environments, *International Symposium on Telecommunication*, 2008 (306-311) NỘI TỐ CHỨC???
- [12] S. M. U.Nor, A. Azizol, and F. A. A.Ahmad, Performance evaluation of AODV, DSDV & DSR routing protocol in grid environment, *IJCSNS International Journal of Computer Science and Network Security* **9** (7) (2009) 261–268.
- [13] Samir R.Das, Charles E. Perkins, Elizabeth M. Royer, “Divison of Computer Science”, The University of Texas at San Antonio, San Antonio, USA; Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks, *Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies* **1** (2000) 3-13. NỘI TỐ CHỨC???
- [14] Tao Lin, “Mobile Ad-hoc Network Routing Protocols: Methodologies and Applications”, Ph.D. in Computer Engineering Thesis, Faculty of the Virginia Polytechnic Institute and State University, Blacksburg, Virginia, March 19, 2004.
- [15] Youssef Iraqi and Raouf Boutaba, “A multi-agent system for resource management in wireless mobile mobile multimedia networks”, University of Waterloo, DSOM 2000 (218–229).

*Ngày nhận bài 30 - 3 - 2012*

*Ngày lại sau sửa ngày 16 - 12 - 2012*