

# MỘT PHƯƠNG PHÁP XÁC ĐỊNH CHI PHÍ MỚI NHẪM CẢI THIỆN CHẤT LƯỢNG DỊCH VỤ ĐỊNH TUYẾN

## A NEW METHOD TO DEFINE THE ROUTING COST FOR IMPROVING QoS

Lương Thái Ngọc<sup>1</sup>, Lê Vũ<sup>2</sup>

<sup>1</sup>Trường Đại học Đồng Tháp; [ltngoc@dthu.edu.vn](mailto:ltngoc@dthu.edu.vn)

<sup>2</sup>Trường Đại học Sư phạm Kỹ thuật - Đại học Đà Nẵng; [levuvn@gmail.com](mailto:levuvn@gmail.com)

**Tóm tắt** - Chi phí định tuyến dựa trên số chặng có ưu điểm là nút nguồn khám phá tuyến ngắn nhất đến đích. Tuy nhiên, hạn chế của phương pháp này là nút nguồn không thể phát hiện nghẽn mạng trong tuyến vừa khám phá, dẫn đến mất gói làm giảm chất lượng dịch vụ (QoS) định tuyến. Bài báo trình bày phương pháp xác định chi phí mới, thay vì sử dụng số chặng (HC), nhóm tác giả dựa vào khả năng tải (LA) của bộ định tuyến là tiêu chí để thiết lập chi phí. Phương pháp này cho phép nút nguồn khám phá ra tuyến có khả năng tải tốt nhất đến đích nhằm giảm thiểu mất gói do nghẽn mạng, ngoài ra nút nguồn có thể phát hiện ra tuyến vừa khám phá bị quá tải hoặc không để có phương án định tuyến phù hợp. Sử dụng NS2, nhóm tác giả đánh giá hiệu quả của hai phương pháp xác định chi phí trong mô hình mạng tải cao sử dụng giao thức AODV. Kết quả cho thấy, chi phí định tuyến sử dụng khả năng tải có tỷ lệ gói tin gửi thành công đến đích lớn hơn khi sử dụng số chặng.

**Từ khóa** - AODV; MANET; HC; LA; QoS

### 1. Giới thiệu

Ngày nay, với sự phát triển bùng nổ của các ứng dụng đa phương tiện truyền thông trên mạng Internet trong khi hạ tầng mạng vẫn chưa đáp ứng được đã tạo ra tình trạng nghẽn mạng làm giảm chất lượng dịch vụ định tuyến. Thời gian qua, các nhà khoa học đã không ngừng nghiên cứu giải pháp phát hiện, hạn chế nghẽn mạng để quá trình truyền thông được thông suốt. Hướng tiếp cận đầu tiên là cải tiến giao thức truyền thông tại tầng vận chuyển là TCP, một số giao thức cải tiến đã được đề xuất như: TCP NewReno [1], Vegas [2], Vegas-W [3]. Hướng tiếp cận khác là cải tiến cơ chế quản lý hàng đợi theo hướng tích cực tại các nút mạng có thể xuất hiện nghẽn [4], một số cải tiến tiêu biểu như: RED [5], ARED [6], FRED [7], REM [8], BLUE [9]. Tuy nhiên, cả hai hướng nghiên cứu này còn tồn tại hạn chế. Ở hướng tiếp cận đầu tiên có hạn chế là tập trung vào việc giải quyết tắc nghẽn mạng khi nó đã hoặc sắp xảy ra dựa trên giao thức TCP, trong khi các luồng dữ liệu đa phương tiện được truyền thông dựa vào giao thức UDP không được quan tâm đến. Ngoài ra, hướng tiếp cận thứ hai có hạn chế là dựa trên xác suất hủy gói sớm ngẫu nhiên dẫn đến mất gói không cần thiết, và chỉ hiệu quả trong mô hình mạng cố định, nơi mà các “nút thắt cổ chai” được xác định trước, chúng không hiệu quả trong các mô hình mạng di động với công nghệ mới như MANET.

Nhóm tác giả nhận thấy rằng, ngoài những nguyên nhân dẫn đến tình trạng nghẽn mạng như: lưu lượng mạng, băng thông và khả năng xử lý của nút. Một nguyên nhân quan trọng khác là do các giao thức định tuyến sử dụng cách tính chi phí dựa vào số chặng. Thuật toán tìm đường theo số chặng chưa phải là thuật toán tốt nhất. Tuyến ngắn nhất có xu hướng đi qua tâm của mạng gây tắc nghẽn cục bộ ở các nút phân bố gần tâm. Vì vậy, cần cải tiến cơ chế tìm đường

**Abstract** - The routing cost determining method based on hop count (HC) has the advantage that it is the source node of the shortest route. However, this method has one drawback that it is impossible for the root node to detect network congestion in the discovered route, which leads to the deterioration of quality of routing service. This article proposes a new routing cost determining method in which load ability (LA) of the routers is used as metric instead of hop count. This method allows source node to discover the route with best loading capacity to minimize the number of lost packages due to network congestion. Furthermore, root node can also determine whether the discovered route is overloaded or not to choose the appropriate routing method. Using NS2, we analyze the effectivity of the two routing cost determining methods in highly loaded network topology using AODV protocol. The results show that LA-based method has higher packet delivery ratio than HC-based method.

**Key words** - AODV; MANET; HC; LA; QoS

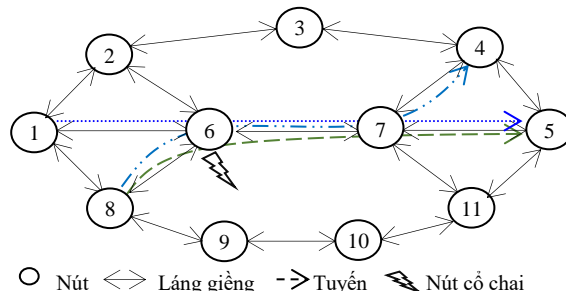
của các giao thức này nhằm giảm tắc nghẽn bởi các lưu lượng bị tập trung tại vùng trung tâm [10, tr. 2].

Bài báo này sử dụng một hướng tiếp cận khác để xác định chi phí định tuyến, cho phép nút nguồn phát hiện nghẽn mạng ngay tại quá trình khám phá tuyến, chi tiết được trình bày trong phần tiếp theo. Phần 3 trình bày quá trình cài đặt giao thức cải tiến từ AODV sử dụng chi phí định tuyến mới. Phần 4 trình bày tham số xây dựng kịch bản mô phỏng và đánh giá kết quả mô phỏng trên NS2 và cuối cùng là kết luận.

### 2. Phương pháp xác định chi phí định tuyến dựa vào khả năng tải

Phần này, trình bày hạn chế của chi phí định tuyến dựa trên số chặng và phương pháp xác định chi phí định tuyến mới dựa vào khả năng tải của bộ định tuyến.

#### 2.1. Hạn chế của chi phí dựa vào số chặng



**Hình 1.** Mô tả kết quả khám phá tuyến sử dụng số chặng

Chi phí định tuyến dựa trên số chặng là số lượng nút mạng từ nguồn đến đích. Một tuyến được xác định là tốt nhất nếu tuyến có số lượng nút đến đích là nhỏ nhất [11]. Hình 1 là ví dụ mô tả quá trình khám phá tuyến của giao thức sử dụng số chặng (tiêu biểu là AODV [12], DSR [13], DSDV [14]) để

tính chi phí định tuyến dẫn đến “nút thắt cổ chai”. Giả sử nút nguồn  $N_1$  khám phá tuyến đến nút đích là  $N_5$ , tương tự nút nguồn  $N_8$  cũng khám phá hai tuyến đến đích là  $N_4$  và  $N_5$ . Kết quả khám phá tuyến với chi phí dựa vào HC sẽ cho ra ba tuyến có chi phí nhỏ nhất là 3, bao gồm:  $\{N_1 \rightarrow N_6 \rightarrow N_7 \rightarrow N_5\}$ ,  $\{N_8 \rightarrow N_6 \rightarrow N_7 \rightarrow N_5\}$ , và  $\{N_8 \rightarrow N_6 \rightarrow N_7 \rightarrow N_4\}$ . Cả ba tuyến giao nhau ở nút cổ chai  $N_6$  dẫn đến cho lưu lượng tải tăng cao và rớt gói tin tại  $N_6$ . Khuyết điểm này có thể khắc phục bằng cách chuyển hướng tuyến đường của nút nguồn qua các bộ định tuyến đang rỗi. Đây chính là vấn đề mà nhóm tác giả phải giải quyết trong bài báo này.

**2.2. Chi phí định tuyến dựa vào khả năng tải**

Thay vì sử dụng tham số HC để xác định chi phí, nhóm tác giả sử dụng khả năng tải của bộ định tuyến là tiêu chí để thiết lập chi phí đến đích. Điều này cho phép nút nguồn phát hiện ra tuyến vừa khám phá có bị quá tải hoặc không hề có phương án định tuyến phù hợp nhằm hạn chế nghẽn mạng.

**a) Định nghĩa 1:** Giả sử tất cả bộ định tuyến có kích thước hàng đợi ( $Q_{max}$ ) là như nhau, khả năng tải ( $LA_{Ri}$ ) của bộ định tuyến tuyến  $R_i$  được xác định như công thức 1.

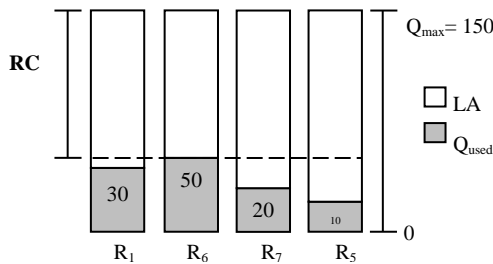
$$LA_{Ri} = (Q_{max} - Q_{used}^{Ri}) / Q_{max} \tag{1}$$

Trong đó,  $Q_{used}^{Ri}$  là kích thước hàng đợi đã được sử dụng tại bộ định tuyến  $R_i$ .

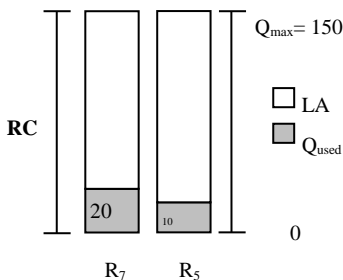
**b) Định nghĩa 2:** Giả sử ta có một tuyến gồm các bộ định tuyến  $R_1 \rightarrow R_2 \rightarrow \dots \rightarrow R_{n-1} \rightarrow R_n$ . Chi phí định tuyến (RC) dựa trên khả năng tải từ nút nguồn  $N_S$  đến đích  $N_D$  là đại lượng đo khả năng tải tối đa tại tất cả các nút trung gian trên tuyến từ  $R_1$  đến  $R_n$  như công thức 2.

$$RC = \min(LA_{Ri}); \forall i = 2..n-1 \tag{2}$$

**Ví dụ 1:** Cho giá trị kích thước hàng đợi đã sử dụng của tất cả các nút như Hình 2, chi phí định tuyến của tuyến  $\{R_1 \rightarrow R_6 \rightarrow R_7 \rightarrow R_5\}$  là  $RC = \text{Min}(0,67; 0,87) = 0,67$ .



Hình 2. Mô tả chi phí định tuyến dựa vào LA



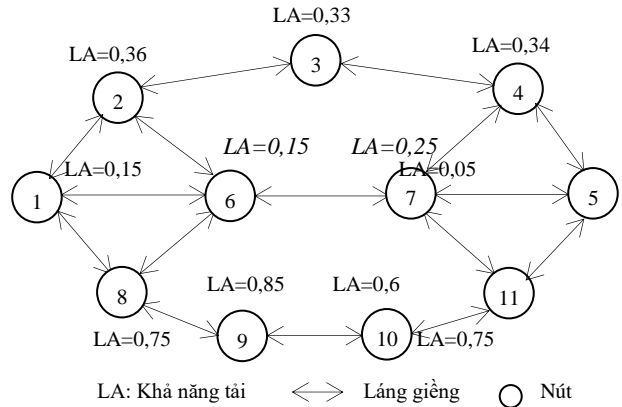
Hình 3. Mô tả chi phí định tuyến của tuyến không có bộ định tuyến trung gian

**Ví dụ 2:** Trong trường hợp tuyến gồm hai bộ định tuyến

là láng giềng với nhau (không có bộ định tuyến trung gian) như Hình 3, chi phí định tuyến của tuyến  $\{R_7 \rightarrow R_5\}$  là  $RC = 1$  vì không có bộ định tuyến trung gian chịu tải giữa nút nguồn và đích.

**Ví dụ 3:** Giả sử tại thời điểm nút nguồn  $N_1$  khám phá tuyến đến đích  $N_5$  thì khả năng tải của các bộ định tuyến như Hình 4. Nút nguồn  $N_1$  có thể đi đến đích qua các tuyến như sau:

- Path<sub>1</sub>:  $N_1 \rightarrow N_2 \rightarrow N_3 \rightarrow N_4 \rightarrow N_5$ ;  $RC_{Path1}=0,33$ ;
- Path<sub>2</sub>:  $N_1 \rightarrow N_6 \rightarrow N_7 \rightarrow N_5$ ;  $RC_{Path2}=0,15$ ;
- Path<sub>3</sub>:  $N_1 \rightarrow N_8 \rightarrow N_9 \rightarrow N_{10} \rightarrow N_{11} \rightarrow N_5$ ;  $RC_{Path3}=0,6$ .



Hình 4. Mô tả tính chi phí định tuyến dựa vào khả năng tải trong mô hình mạng tổng quát

Như vậy, tuyến được chọn là tuyến  $\{N_1 \rightarrow N_8 \rightarrow N_9 \rightarrow N_{10} \rightarrow N_{11} \rightarrow N_5\}$  vì Path<sub>3</sub> có khả năng tải tốt nhất, tương ứng với chi phí tốt nhất là  $RC=0,6$ . Trong trường hợp có nhiều tuyến đến đích thì tuyến được chọn là tuyến có khả năng tải lớn nhất, tương ứng có chi phí RC lớn nhất. Trong trường hợp tồn tại hai tuyến đến đích có chi phí tốt nhất bằng nhau thì tuyến có số chặng thấp hơn sẽ ưu tiên được chọn.

**Ví dụ 4:** Xét lại mô hình mạng như Hình 4, trong trường hợp hệ thống bắt đầu vận hành thì tất cả các bộ định tuyến đều rỗi ( $LA=0$ ), dẫn đến tất cả tuyến đường từ  $N_1$  đến đích  $N_5$  đều có chi phí bằng nhau ( $RC=1$ ). Kết quả là tuyến Path<sub>2</sub> được chọn vì có số lượng nút đến đích thấp nhất, lúc này kết quả khám phá tuyến trùng với cách thức tính chi phí dựa trên số chặng.

**3. LA-AODV - Giao thức định tuyến cải tiến sử dụng chi phí định tuyến dựa trên khả năng tải**

Phần này trình bày cơ chế khám phá tuyến của giao thức AODV sử dụng chi phí dựa trên HC và chi tiết quá trình khám phá tuyến của giao thức LA-AODV sử dụng chi phí định tuyến mới.

**3.1. Giao thức định tuyến AODV**

AODV thuộc nhóm giao thức định tuyến theo yêu cầu, khám phá tuyến thông qua gói yêu cầu tuyến (RREQ) và gói trả lời tuyến (RREP). Cấu trúc gói tin điều khiển tuyến gồm RREQ và RREP của giao thức AODV sử dụng chi phí định tuyến dựa vào HC như Hình 5.

Khi nút nguồn  $N_S$  muốn gửi thông tin đến nút đích  $N_D$  mà không có tuyến đến đích trong bảng định tuyến của nó,  $N_S$  tiến hành khám phá tuyến bằng cách phát quảng bá gói RREQ đến các nút láng giềng của nó. Nút trung gian  $N_i$  lưu tuyến ngược

về nguồn và tiếp tục quảng bá gói RREQ đến tất cả láng giềng. Quá trình lưu tuyến ngược về nguồn và quảng bá gói RREQ tiếp tục cho đến khi nút đích  $N_D$  nhận được gói yêu cầu tuyến hoặc nút đích không tồn tại. Để tránh xử lý trùng lặp, mỗi gói RREQ chỉ được nút trung gian xử lý một lần.

Type	J	R	C	Reserved	HC
Broadcast ID					
Destination IP Address					
Destination Sequence Number					
Source IP Address					
Source Sequence Number					

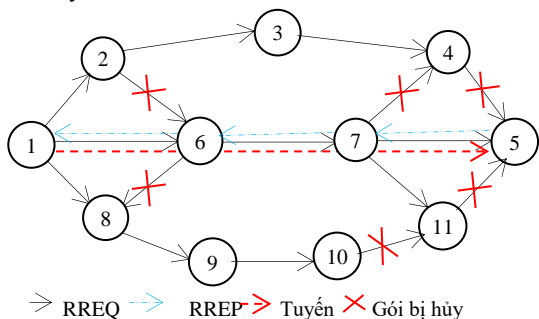
a) RREQ packet

Type	R	A	Reserved	Prefix Sz	HC
Destination IP Address					
Destination Sequence Number					
Source IP Address					
Lifetime					

b) RREP packet

**Hình 5.** Gói tin điều khiển tuyến của giao thức AODV sử dụng chi phí định tuyến dựa vào HC [12]

Khi nhận được gói RREQ, nút đích  $N_D$  trả lời tuyến thông qua gói RREP chứa thông tin tuyến về nguồn. Các nút trung gian chuyển tiếp gói RREP về nguồn thông qua tuyến ngược đã lưu khi nhận gói RREQ trước đó, đồng thời lưu tuyến đến đích  $N_D$  vào bảng định tuyến của nó trước khi chuyển tiếp gói RREP về nguồn. Ngoài ra, tại các nút trung gian cũng thực hiện quá trình trả lời tuyến RREP nếu tồn tại tuyến đủ “tươi” đến đích.



**Hình 6.** Khám phá tuyến của giao thức AODV

Xem mô hình mạng tại Hình 6, nút nguồn  $N_1$  khám phá tuyến đến nút đích  $N_5$  bằng cách phát quảng bá gói RREQ đến các láng giềng  $\{N_2, N_6, N_7\}$ , gói RREQ được khởi tạo giá trị là  $[N_1, N_5, 0]$ . Khi nhận được gói yêu cầu tuyến, nút  $N_2$  kiểm tra và thấy rằng nó không là nút đích nên tăng HC lên 1 và tiếp tục quảng bá gói RREQ đến tất cả láng giềng của nó gồm  $\{N_3, N_6\}$ , đồng thời lưu tuyến ngược về nguồn  $N_1$ , quá trình tiếp tục tại nút  $N_6, N_7$  và các nút trung gian khác cho đến khi nút  $N_5$  nhận được RREQ.

Khi nhận được gói RREQ từ nút  $N_7$ , nút đích  $N_5$  trả lời gói RREP về nguồn thông qua nút trung gian  $N_7$ , gói RREP được khởi tạo giá trị là  $[N_5, N_1, 0]$ .  $N_7$  kiểm tra và thấy rằng nó không phải đích đến của gói RREP (không phải nút nguồn) nên tăng HC lên 1 và tiếp tục chuyển tiếp về nguồn thông qua nút trung gian  $N_6$ . Kết quả là  $N_1$  nhận được gói RREP thông qua nút trung gian  $N_6$  với HC bằng 3. Bảng 1 mô tả thông tin tuyến đến đích và nguồn tại tất cả các nút

sau quá trình khám phá tuyến. Kết quả là nút nguồn  $N_1$  khám phá ra tuyến ngắn nhất đến đích  $N_5$  theo tuyến  $\{N_1 \rightarrow N_6 \rightarrow N_7 \rightarrow N_5\}$  với chi phí HC là 3.

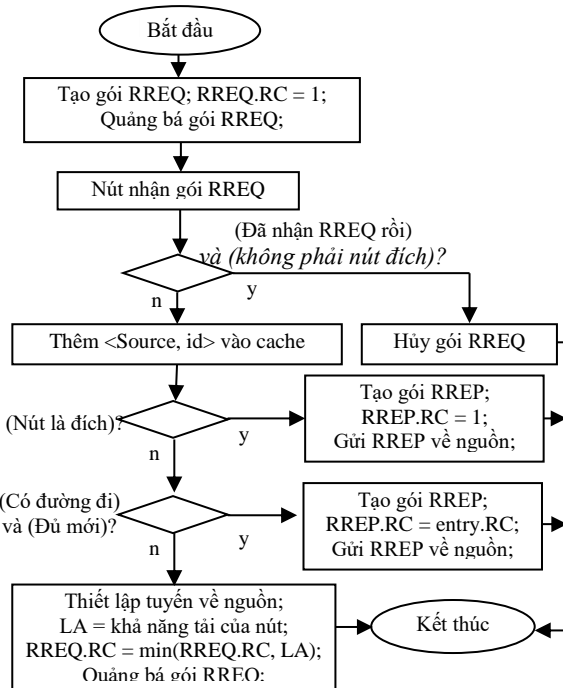
**Bảng 1.** Kết quả khám phá tuyến của AODV với chi phí định tuyến dựa trên HC; S: Nguồn, D: Đích, N: Nút, NH: Nút kế, HC: Chi phí định tuyến

Bước	Nút	RREQ/ RREP [S, D, HC]	Bảng định tuyến (RT)		
			N	NH	HC
Quảng bá RREQ	$N_1$	Khởi tạo gói RREQ $[N_1, N_5, 0]$			
	$N_2$	$[N_1, N_5, 1]$	$N_1$	$N_1$	1
	$N_3$	$[N_1, N_5, 2]$	$N_1$	$N_2$	2
	$N_4$	$[N_1, N_5, 3]$	$N_1$	$N_3$	3
	$N_5$	$[N_1, N_5, 3]$	$N_1$	$N_7$	3
	$N_6$	$[N_1, N_5, 1]$	$N_1$	$N_1$	1
	$N_7$	$[N_1, N_5, 2]$	$N_1$	$N_6$	2
	$N_8$	$[N_1, N_5, 1]$	$N_1$	$N_1$	1
	$N_9$	$[N_1, N_5, 2]$	$N_1$	$N_8$	2
	$N_{10}$	$[N_1, N_5, 3]$	$N_1$	$N_9$	3
	$N_{11}$	$[N_1, N_5, 3]$	$N_1$	$N_7$	3
Gửi gói RREP	$N_5$	Khởi tạo gói RREP $[N_5, N_1, 0]$			
	$N_7$	$[N_5, N_1, 1]$	$N_5$	$N_5$	1
	$N_6$	$[N_5, N_1, 2]$	$N_5$	$N_7$	2
	$N_1$	$[N_5, N_1, 3]$	$N_5$	$N_6$	3

(\*) Tuyến vừa khám phá

Tương tự, kết quả khám phá tuyến của nút nguồn  $N_8$  đến hai nút đích  $N_4$  và  $N_5$  cũng thông qua NH là  $N_6$  với chi phí là 3. Ta thấy rằng trong 3 tuyến vừa khám phá giao nhau ở “nút thắt cổ chai” là  $N_6$ .

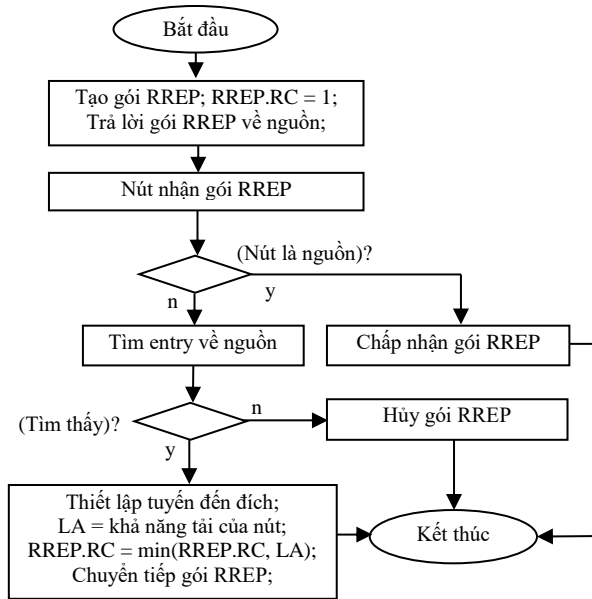
**3.2. Giao thức cải tiến LA-AODV**



**Hình 7.** Thuật toán yêu cầu tuyến

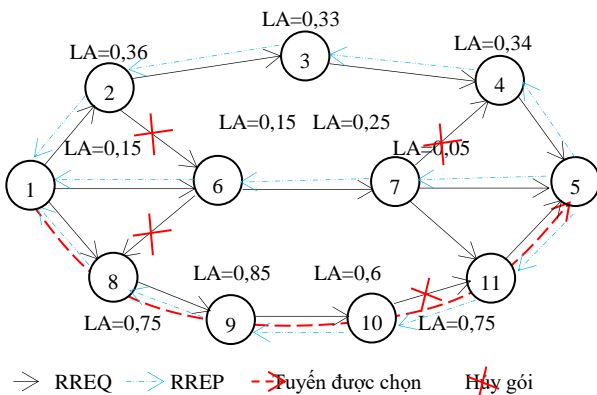
Để cài đặt giao thức LA-AODV, nhóm tác giả thực hiện như sau: *Đầu tiên*, nhóm tác giả thay thế trường HC thành tên mới là RC trong hai gói tin điều khiển tuyến RREQ và RREP để lưu chi phí định tuyến dựa trên khả năng tải; *Tiếp theo*, thay thế thuộc tính HC thành thuộc tính RC của thông tin định tuyến (Entry) trong bảng định tuyến (Routing Table)

để phù hợp với chi phí định tuyến dựa trên khả năng tải; Cuối cùng, cải tiến thuật toán khám phá tuyến của giao thức AODV thành LA-AODV, cho phép nút nguồn khám phá tuyến dựa vào khả năng tải. Hình 7 và 8 trình bày lưu đồ thuật toán của giao thức cải tiến LA-AODV.



Hình 8. Thuật toán trả lời tuyến

**Ví dụ minh họa:** Giả sử tại thời điểm khám phá tuyến thì khả năng tải của các bộ định tuyến như Hình 9. Nút nguồn  $N_1$  khám phá tuyến đến nút đích  $N_5$  với chi phí dựa vào khả năng tải. Gói yêu cầu tuyến được quảng bá đến đích  $N_5$  theo ba hướng gồm  $\{N_1 \rightarrow N_2 \rightarrow N_3 \rightarrow N_4 \rightarrow N_5\}$ ,  $\{N_1 \rightarrow N_6 \rightarrow N_7 \rightarrow N_5\}$ , và  $\{N_1 \rightarrow N_8 \rightarrow N_9 \rightarrow N_{10} \rightarrow N_{11} \rightarrow N_5\}$ . Nút trung gian chỉ xử lý gói RREQ đầu tiên nhận được và lưu thông tin tuyến ngược về nguồn vào bảng định tuyến.



Hình 9. Khám phá tuyến của giao thức AODV sử dụng gói RREQ và RREP, chi phí dựa trên RC

Khi nhận gói tin yêu cầu tuyến, nút đích  $N_5$  trả lời ba gói RREP về nguồn trên ba tuyến theo các hướng gồm  $\{N_5 \rightarrow N_4 \rightarrow N_3 \rightarrow N_2 \rightarrow N_1\}$ ,  $\{N_5 \rightarrow N_7 \rightarrow N_6 \rightarrow N_1\}$ , và  $\{N_5 \rightarrow N_{11} \rightarrow N_{10} \rightarrow N_9 \rightarrow N_8 \rightarrow N_1\}$ . Do vậy, nút nguồn nhận được ba gói RREP lần lượt đến từ các nút  $N_6, N_2$  và  $N_8$ . Gói RREP đầu tiên nhận được từ  $N_2$  được sử dụng để thêm tuyến mới đến đích  $N_5$  thông qua NH là  $N_6$  với chi phí  $RC = \min(0,15; 0,25) = 0,15$ . Gói RREP thứ hai đến từ  $N_2$  có chi phí  $RC = \min(0,36; 0,33; 0,34) = 0,33$ . Vì vậy,  $N_1$

cập nhật lại tuyến đến đích thông qua NH là  $N_2$  vì có chi phí tốt hơn tuyến hiện tại (tuyến đã lưu khi nhận gói RREP đầu tiên). Cuối cùng,  $N_1$  nhận được gói RREP thứ ba đến từ  $N_8$  có chi phí  $RC = \min(0,75; 0,85; 0,6; 0,75) = 0,6$ . Đây là tuyến tốt nhất nên  $N_1$  tiếp tục cập nhật lại thông tin tuyến đến đích thông qua NH là  $N_8$  với chi phí là 0,6.

Bảng 2 cho thấy thông tin gói RREQ, RREP và chi tiết thông tin tuyến được thiết lập tại mỗi nút. Ngoài ra, kết quả khám phá tuyến được ghi nhận cho thấy rằng nút nguồn  $N_1$  và nút đích  $N_5$  đã phát hiện ra tuyến  $\{N_5 \rightarrow N_7 \rightarrow N_6 \rightarrow N_1\}$  có khả năng bị nghẽn mạng, do khả năng tải lớn nhất của tuyến chỉ đạt 0,15, tương ứng với hàng đợi của nút cổ chai chỉ còn trống 15%.

Bảng 2. Kết quả khám phá tuyến của AODV với chi phí định tuyến dựa trên RC; S: Nguồn, D: Đích, N: Nút, NH: Nút kế, RC: Chi phí định tuyến

Bước	Nút	RREQ/ RREP [S, D, RC]	Bảng định tuyến (RT)			
			N	NH	RC	
Quảng bá RREQ	$N_1$	Khởi tạo gói RREQ [ $N_1, N_5, 1$ ]				
	$N_2$	[ $N_1, N_5, 0,36$ ]	$N_1$	$N_1$	1	
	$N_3$	[ $N_1, N_5, 0,33$ ]	$N_1$	$N_2$	0,36	
	$N_4$	[ $N_1, N_5, 0,33$ ]	$N_1$	$N_3$	0,33	
	$N_5$	[ $N_1, N_5, 0,33$ ]	$N_1$	$N_7$	0,33	
	$N_6$	[ $N_1, N_5, 0,15$ ]	$N_1$	$N_1$	1	
	$N_7$	[ $N_1, N_5, 0,15$ ]	$N_1$	$N_6$	0,15	*
	$N_8$	[ $N_1, N_5, 0,05$ ]	$N_1$	$N_7$	0,15	*
	$N_9$	[ $N_1, N_5, 0,75$ ]	$N_1$	$N_1$	1	
	$N_{10}$	[ $N_1, N_5, 0,6$ ]	$N_1$	$N_9$	0,75	
	$N_{11}$	[ $N_1, N_5, 0,6$ ]	$N_1$	$N_7$	0,6	
Trả lời gói RREP	$N_5$	Khởi tạo gói RREP [ $N_5, N_1, 1$ ] thứ nhất				
	$N_7$	[ $N_5, N_1, 0,25$ ]	$N_5$	$N_5$	1	
	$N_6$	[ $N_5, N_1, 0,15$ ]	$N_5$	$N_7$	0,25	
	$N_1$	[ $N_5, N_1, 0,15$ ]	$N_5$	$N_6$	0,15	*
	$N_5$	Khởi tạo gói RREP [ $N_5, N_1, 1$ ] thứ hai				
	$N_4$	[ $N_5, N_1, 0,34$ ]	$N_5$	$N_5$	1	
	$N_3$	[ $N_5, N_1, 0,33$ ]	$N_5$	$N_4$	0,34	
	$N_2$	[ $N_5, N_1, 0,33$ ]	$N_5$	$N_3$	0,33	
	$N_1$	[ $N_5, N_1, 0,15$ ]	$N_5$	$N_2$	0,33	
	$N_5$	Khởi tạo gói RREP [ $N_5, N_1, 1$ ] thứ ba				
	$N_{11}$	[ $N_5, N_1, 0,75$ ]	$N_5$	$N_5$	1	
$N_{10}$	[ $N_5, N_1, 0,6$ ]	$N_5$	$N_{11}$	0,75		
$N_9$	[ $N_5, N_1, 0,6$ ]	$N_5$	$N_{10}$	0,6		
$N_8$	[ $N_5, N_1, 0,6$ ]	$N_5$	$N_9$	0,6		
$N_1$	[ $N_5, N_1, 0,15$ ]	$N_5$	$N_8$	0,6		

3.3. So sánh AODV và LA-AODV

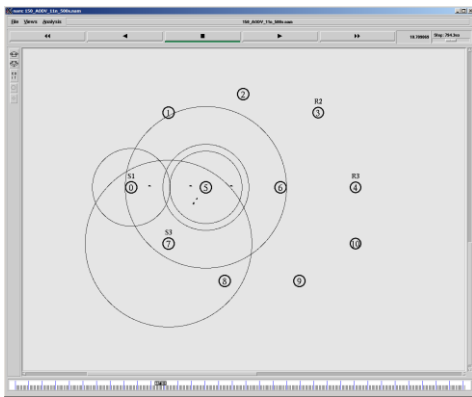
Đặc điểm của giao thức cải tiến được nhóm tác giả đánh giá so với giao thức gốc dựa trên một số tiêu chí như Bảng 3. Giao thức LA-AODV khám phá ra tuyến có khả năng thông qua lớn để hạn chế nghẽn. Trong quá trình khám phá tuyến, LA-AODV có khả năng phát hiện nghẽn mạng nên hoạt động hiệu quả trong môi trường mạng tải cao.

**Bảng 3.** So sánh hai giao thức AODV và LA-AODV

Tiêu chí	AODV	LA-AODV
Chi phí định tuyến dựa vào	HC	LA
Tuyến tốt nhất là tuyến chi phí	Nhỏ	Lớn
Khả năng xuất hiện nút cổ chai	Cao	Thấp
Phát hiện nghẽn mạng	Không	Có
Nút đích xử lý gói RREQ	Đầu tiên	Tất cả
Hiệu quả khi lưu lượng tải cao	Không	Tốt

**4. Mô phỏng đánh giá kết quả**

Nhóm tác giả sử dụng hệ mô phỏng NS2 phiên bản 2.35 để đánh giá hiệu quả của chi phí định tuyến dựa vào LA so với chi phí định tuyến dựa vào HC. Mô hình mạng có 11 nút hoạt động trong phạm vi 2.000m x 2.000m, các nút mạng bố trí cố định như Hình 9, giao diện mô phỏng trên NS2 như Hình 10.



**Hình 10.** Giao diện mô phỏng trên NS2

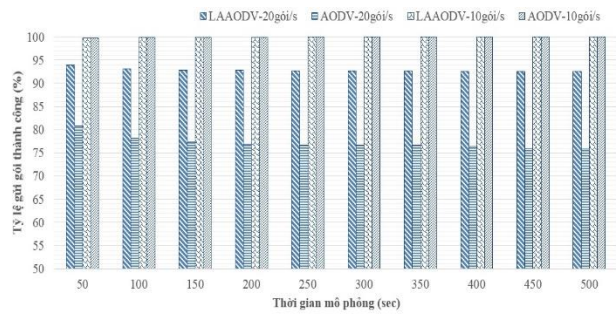
Bảng 4 mô tả chi tiết thông số mô phỏng, nhóm tác giả sử dụng ba luồng dữ liệu CBR như mô tả tại Hình 7, luồng đầu tiên từ nút nguồn N<sub>0</sub> đến đích N<sub>4</sub> bắt đầu phát từ giây thứ 0, luồng thứ hai từ nút nguồn N<sub>7</sub> đến đích N<sub>3</sub> bắt đầu phát từ giây thứ 10, luồng cuối cùng từ nút nguồn N<sub>7</sub> đến đích N<sub>4</sub> bắt đầu phát từ giây thứ 20. Tốc độ phát lần lượt là 10 gói/giây hoặc 20 gói/giây.

**Bảng 4.** Thông số mô phỏng

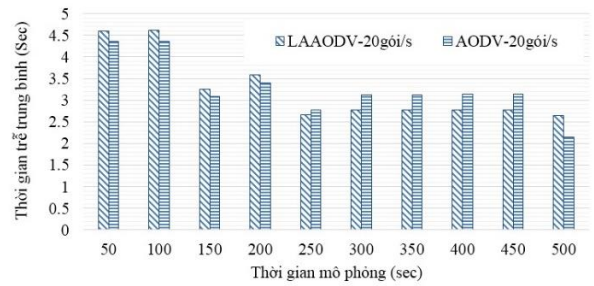
Thông số	Giá trị
Khu vực địa lý	2.000 m x 2000 m
Vùng thu phát sóng	250 m
Thời gian mô phỏng	1.000 s
Tổng số nút mạng	11
Dạng truyền thông	CBR (Constant Bit Rate)
Số kết nối	3 UDP
Tốc độ phát	10 gói/giây; 20 gói/giây
Kích thước gói tin	512(bytes)
Hàng đợi	FIFO (DropTail)
Giao thức định tuyến	AODV và LA-AODV

Kết quả mô phỏng tại Hình 11 cho thấy rằng tỷ lệ gửi gói thành công (PDR) của LA-AODV tương đương với giao thức gốc. Tuy nhiên, trong môi trường tải cao 20 gói/s thì tỷ lệ gửi gói thành công của LA-AODV hiệu quả hơn giao thức gốc. Kết thúc 500s mô phỏng thì PDR của LA-AODV đạt

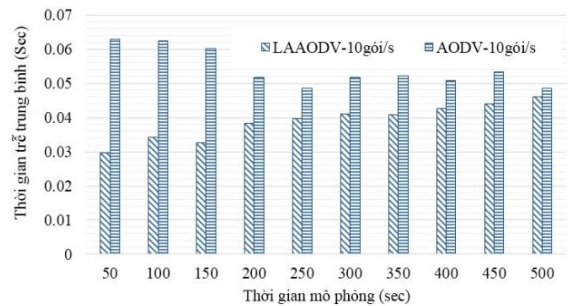
92,59%, cao hơn 16,7% so với AODV (đạt 75,87%).



**Hình 11.** Tỷ lệ gửi gói thành công



a) Tải cao



b) Tải thấp

**Hình 12.** Thời gian trễ trung bình

Thời gian trễ trung bình tại Hình 12 cho thấy rằng giao thức AODV trong môi trường tải thấp thì thời gian trễ trung bình (ETE) của LA-AODV thấp hơn AODV. Tuy nhiên, trong môi trường tải cao thì ETE của LA-AODV cao hơn AODV do tuyến đường khám phá dài hơn (dựa trên hop) AODV. Kết thúc 500s mô phỏng thì ETE của LAAODV là 2,64s và AODV là 2,15s trong môi trường tải cao; tương ứng là 0,046s và 0,049s trong môi trường tải thấp.

**5. Kết luận**

Như vậy, nhóm tác giả đã đề xuất một cơ chế xác định chi phí định tuyến mới cho giao thức AODV trên mạng MANET. Việc thiết lập chi phí định tuyến dựa trên khả năng tải cho phép khám phá ra tuyến đường với khả năng tải cao, hạn chế tắc nghẽn. Kết quả mô phỏng trên NS2 đã cho thấy hiệu quả của giải pháp đề xuất. Tương lai, nhóm tác giả sẽ tiếp tục nghiên cứu, cải tiến và mô phỏng trong nhiều môi trường mạng khác nhau để đánh giá hiệu quả.

**Cảm ơn:** Bài báo được sự hỗ trợ tài chính của Quỹ Phát triển Khoa học và Công nghệ, Đại học Đà Nẵng, mã số B2016-DNA-46-TT.

## Phụ lục

Giao thức LA-AODV được cải tiến từ AODV nên nhóm tác giả sử dụng mã lệnh của giao thức AODV có sẵn trên NS2 (\ns-allinone-2.35\ns-2.35\aodv) để cài đặt. Sau đây là một số hiệu chỉnh cơ bản để cài đặt:

Đầu tiên, thay đổi cấu trúc của gói RREQ tập tin *laaodv\_packet.h* bằng cách loại bỏ thuộc tính *rq\_hop\_count* và bổ sung vào thuộc tính *rq\_la*.

```

74 struct hdr_laaodv_request {
75     u_int8_t    rq_type;        // Packet Type
76     u_int8_t    reserved[2];
77     float       rq_la;         // Load Ability
78     u_int32_t   rq_bcast_id;   // Broadcast ID
79
80     nsaddr_t    rq_dst;        // Destination IP Address
81     u_int32_t   rq_dst_seqno;  // Destination Sequence Number
82     nsaddr_t    rq_src;        // Source IP Address
83     u_int32_t   rq_src_seqno;  // Source Sequence Number
84
85     double      rq_timestamp;  // when REQUEST sent;
86     // used to compute route discovery latency

```

Ngoài ra, thông tin định tuyến của giao thức mới chứa RC nên cấu trúc của gói RREP bị loại bỏ thuộc tính *rp\_hop\_count* và bổ sung vào thuộc tính *rp\_la*.

```

111 struct hdr_laaodv_reply {
112     u_int8_t    rp_type;        // Packet Type
113     u_int8_t    reserved[2];
114     float       rp_la;         // Load Ability
115     nsaddr_t    rp_dst;        // Destination IP Address
116     u_int32_t   rp_dst_seqno;  // Destination Sequence Number
117     nsaddr_t    rp_src;        // Source IP Address
118     double      rp_lifetime;   // Lifetime
119
120     double      rp_timestamp;  // when corresponding REQ sent;
121     // used to compute route discovery latency

```

Tiếp theo, thay đổi cấu trúc của bảng định tuyến để lưu trữ chỉ phí định tuyến LA. Mở tập tin *laaodv\_rtable.h*, loại bỏ thuộc tính *rt\_hops*, thay bằng thuộc tính *rt\_la*.

```

81 class laaodv_rt_entry {
82     friend class laaodv_rtable;
83     friend class LAODV;
84     friend class LALocalRepairTimer;
85 public:
86     laaodv_rt_entry();
87     ~laaodv_rt_entry();
88
89     void    nb_insert(nsaddr_t id);
90     LAODV_Neighbor* nb_lookup(nsaddr_t id);
91
92     void    pc_insert(nsaddr_t id);
93     LAODV_Precursor* pc_lookup(nsaddr_t id);
94     void    pc_delete(nsaddr_t id);
95     void    pc_delete(void);
96     bool    pc_empty(void);
97
98     double  rt_req_timeout;    // when I can send another req
99     u_int8_t  rt_req_cnt;      // number of route requests
100
101 protected:
102     LIST_ENTRY(laaodv_rt_entry) rt_link;
103
104     nsaddr_t  rt_dst;
105     u_int32_t  rt_seqno;
106     /* u_int8_t  rt_interface; */
107     float     rt_la;          // LA
108     int       rt_last_hop_count; // last valid hop count
109     nsaddr_t  rt_nexthop;    // next hop IP address

```

Tiếp theo, khi gửi gói RREQ thuộc tính *rq\_la* của gói RREQ được thiết lập là 1, thực hiện tương tự cho thuộc tính *rp\_la* gói RREP.

```

1188 // Fill up some more fields.
1189 rq->rq_type = LAODVTYPE_RREQ;
1190 rq->rq_la = 1;
1191 rq->rq_bcast_id = bid++;
1192 rq->rq_dst = dst;
1193 rq->rq_dst_seqno = (rt ? rt->rt_seqno : 0);
1194 rq->rq_src = index;
1195 seqno += 2;
1196 assert ((seqno%2) == 0);
1197 rq->rq_src_seqno = seqno;
1198 rq->rq_timestamp = CURRENT_TIME;
1199
1200
1201 Scheduler::instance().schedule(target_, p, 0.);

```

Tiếp theo, khi nhận gói RREQ nút chỉ loại bỏ gói tin đã xử lý rồi nếu nút hiện tại không phải đích.

```

686
687 if (id_lookup(rq->rq_src, rq->rq_bcast_id) && (rq->rq_dst != index)) {
688
689 #ifdef DEBUG
690     fprintf(stderr, "%s: discarding request\n", __FUNCTION__);
691 #endif // DEBUG
692
693     Packet::free(p);
694     return;
695 }

```

Cuối cùng, trước khi chuyển tiếp gói RREQ, nút cập nhật lại chỉ phí định tuyến trong gói RREQ.

```

826 * Can't reply. So forward the Route Request
827 */
828 else {
829     ih->saddr() = index;
830     ih->daddr() = IP_BROADCAST;
831     double qz = (double) ifqueue->queue_length() / ifqueue->queue_maxlength();
832     rq->rq_la = (rq->rq_la > 1 - qz) ? (1 - qz) : rq->rq_la; // LA
833     // Maximum sequence number seen en route
834     if (rt) rq->rq_dst_seqno = max(rt->rt_seqno, rq->rq_dst_seqno);
835     forward((Laaodv_rt_entry*) 0, p, DELAY);
836 }

```

## TÀI LIỆU THAM KHẢO

- [1] N. Parvez, A. Mahanti, and C. Williamson, "An analytic throughput model for TCP NewReno", *IEEE/ACM Transactions on Networking*, Vol. 18, No. 2, 2010, pp. 448-461.
- [2] M. Podlesny and C. Williamson, *Providing fairness between TCP NewReno and TCP Vegas with RD network services*, in IEEE International Workshop on Quality of Service, IWQoS, 2010.
- [3] L. Ding, X. Wang, Y. Xu, W. Zhang, and W. Chen, *Vegas-W: An enhanced TCP-Vegas for wireless ad hoc networks*, in IEEE International Conference on Communications, 2008, pp. 2383-2387.
- [4] R. Adams, "Active Queue Management: A Survey", *IEEE Communications Surveys & Tutorials*, Vol. 15, No. 3, 2013, pp. 1425-1476.
- [5] S. Patel, *Performance analysis of RED for stabilized queue*, in 2014 7th International Conference on Contemporary Computing, IC3 2014, 2014, pp. 306-311.
- [6] J. Chen, C. Hu, and Z. Ji, "An improved ARED algorithm for congestion control of network transmission", *Mathematical Problems in Engineering*, Vol. 2010, 2010.
- [7] N. Glombitza, M. Lipphardt, H. Hellbruck, and S. Fischer, *FRED - An Application for a Real-Life Large Scale Multihop Ad Hoc Network*, in 2008 Fifth Annual Conference on Wireless on Demand Network Systems and Services, 2008, pp. 73-76.
- [8] S. Athuraliya, S. H. Low, and V. H. Li, "REM: Active queue management", *IEEE Network*, Vol. 15, No. 3, 2001, pp. 48-53.
- [9] W. Feng, D. D. Kandlur, D. Saha, and K. G. Shin, *BLUE: A new class of active queue management algorithms*, Ann Arbor, 1999, pp. 1-27.
- [10] Đ. Đ. Cường, N. V. Tam, and N. G. Hiểu, *Nghiên cứu cải tiến hiệu năng giao thức định tuyến AODV và AOMDV trong mạng MANET*, Luận án tiến sĩ, Học viện Khoa học và Công nghệ, 2017.
- [11] E. Alotaibi and B. Mukherjee, "A survey on routing algorithms for Wireless Ad-hoc and Mesh networks", *Computer Networks*, Vol. 56, No. 2, 2012, pp. 940-965.
- [12] C. E. Perkins, M. Park, and E. M. Royer, *Ad-hoc On-Demand Distance Vector Routing*, in Proceedings of Second IEEE Workshop on Mobile Computing Systems and Applications (WMCSA), 1999, pp. 90-100.
- [13] D. B. Johnson and D. A. Maltz, *Dynamic Source Routing in Ad Hoc Wireless Networks*, Mobile Computing, The Kluwer International Series in Engineering and Computer Science, Vol. 353, Springer, 1996, pp. 153-181.
- [14] C. E. Perkins and P. Bhagwat, *Highly Dynamic Destination Sequence Distance Vector (DSDV) for Mobile Computers*, in Proceedings of the SIGCOMM 1994 Conference on Communications Architectures, Protocols and Applications, 1994, pp. 234-244.