DOI: 10.15625/vap.2023.0107

RLMR: MỘT PHƯƠNG PHÁP ÁP DỤNG Q-LEARNING CHO ĐỊNH TUYẾN TRONG MẠNG TÙY BIẾN DI ĐỘNG

Nguyễn Quốc Cường[1,2], Mai Cường Thọ[1,3], Lê Hữu Bình[1], Võ Thanh Tú[1]

[1]Khoa Công nghệ thông tin, Trường Đại học Khoa học, Đại học Huế
[2]Khoa Khoa học Tự nhiên và Công nghệ, Trường Đại học Tây Nguyên
[3]Khoa Công nghệ thông tin, Trường Đại học Nha Trang

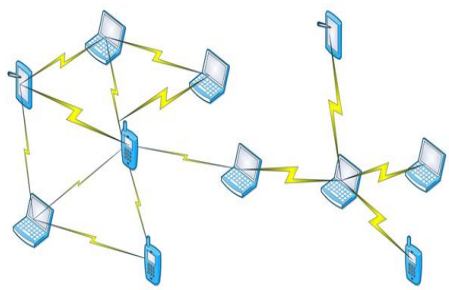
nqcuong.dhkh23@hueuni.edu.vn, mctho@hueuni.edu.vn, lhbinh@hueuni.edu.vn, vttu@hueuni.edu.vn

TÓM TẮT: *Hiện nay, việc nghiên cứu học tăng cường cho các giao thức định tuyến đã nhận được nhiều sự quan tâm. Trong MANET, tính di động cao của các nút dẫn đến cấu trúc liên kết động và không bền vững, đó là thách thức yêu cầu giao thức định tuyến cần phải thích ứng nhanh. Học tăng cường đã được chứng minh là có thể giải quyết được thách thức định tuyến này bằng cách cho các nút mạng quan sát và thu thập thông tin từ môi trường hoạt động cục bộ của chúng, học và đưa ra các quyết định định tuyến một cách hiệu quả. Bài báo này, tập trung vào việc áp dụng mô hình học tăng cường để định tuyến trong mạng MANET nhằm nâng cao hiệu năng mạng. Kết quả mô phỏng sử dụng NS2 cho thấy rằng, giao thức định tuyến có ứng dụng học tăng cường đã nâng cao được hiệu năng về tỉ lệ gửi gói dữ liệu thành công và thông lượng mạng.*

Từ khóa: *MANET, Q-Learning, Giao thức định tuyến, DSDV.*

I. GIỚI THIỆU

MANET (Mobile Ad Hoc Network) [1] là mạng tùy biến di động, trong đó các nút mạng đều có khả năng di chuyển nên không có một nút mạng cố định nào thực hiện chức năng điều khiển trung tâm. Trong mạng này các thiết bị di động không dây kết nối ngang hàng với nhau hình thành nên một mạng tạm thời mà không cần sự trợ giúp của các thiết bị trung tâm cũng như các cơ sở hạ tầng mạng cố định nên nó vừa đóng vai trò truyền nhận vừa đóng vai trò như thiết bị định tuyến (Hình 1).



Hình 1. Ví dụ về mạng MANET

Với các đặc điểm trên, mạng MANET được ứng dụng nhiều trong các lĩnh vực như: hoạt động quân sự-triển khai tác chiến trên chiến trường không có cơ sở hạ tầng mạng cố định; trong hội nghị, sân bay, trường học; hoạt động ứng cứu khẩn cấp (thiên tai sóng thần, động đất); ứng dụng cho các thiết bị thông minh kết nối Internet; ứng dụng trong giao thông - các phương tiện giao thông được gắn thiết bị MANET để truyền thông với nhau.

MANET hoạt động như một mạng ngang hàng không có trung tâm điều khiển, các nút thường xuyên di chuyển nên cấu trúc liên kết cũng thay đổi theo [2]. Vì vậy, bảng định tuyến tại mỗi nút phải được cập nhật thường xuyên để đáp ứng với những thay đổi về cấu trúc liên kết. Do đó, với mong muốn xây dựng một giao thức định tuyến thích ứng và có tính tự trị cao, có nghĩa là giao thức định tuyến của MANET phải có khả năng tìm ra một láng giềng tối ưu nhất để hoàn thành quá trình truyền tin, thông qua việc cảm nhận sự thay đổi của môi trường một cách thích ứng. Sử dụng các kỹ thuật học máy tăng cường trong thiết kế giao thức định tuyến đã được sử dụng nhiều và mang lại hiệu quả tốt trong mạng MANET, do các thuật toán học máy có thể học và thích nghi với môi trường thay đổi [3].

Gần đây, nhiều nhóm nghiên cứu đã áp dụng các kỹ thuật học tăng cường (RL-Reinforcement Learning) để cải thiện các giao thức định tuyến trong mạng MANET [4, 6, 7, 8, 9]. Bằng cách sử dụng RL, các tác giả của [4] đã đề xuất thuật toán định tuyến cho mạng MANET nhằm tăng tuổi thọ của mạng. Thuật toán này sử dụng Q-Learning để lấy thông tin trạng thái toàn cầu từ các liên lạc cục bộ. Các tuyến đường sau đó sẽ được cập nhật dựa trên những gì đã học được. Kết quả mô phỏng cho thấy rằng giao thức đề xuất vượt trội so với AODV - SARSA [5] về tỷ lệ phân phối gói và mức tiêu thụ năng lượng. Cũng sử dụng phương pháp RL, các tác giả của [6] đã đề xuất giao thức định tuyến cân bằng năng lượng dựa trên Q-Learning (QEBR) cho WMN. QEBR được thực thi theo nguyên tắc định tuyến phân tán. Các tác giả đã đề xuất khái niệm phân loại năng lượng lân cận để sử dụng làm phần thưởng cho thuật toán Q-Learning. Công trình [7] đề xuất một thuật toán định tuyến đảm bảo QoS sử dụng học tăng cường cho mạng WMN có tên là QGIR. Trong đó, có xây dựng hàm phần thưởng cho thuật toán Q-Learning để chọn đường đi sao cho tỷ lệ gửi gói tin là cao nhất. Đồng thời, hệ số tỷ lệ học tập được thay đổi linh hoạt để xác định ràng buộc của độ trễ đầu - cuối. Tại mỗi nút có cơ chế tự học thông tin định tuyến tại theo từng chu kỳ cập nhật tải lưu lượng. Trong nghiên cứu [8], sử dụng độ

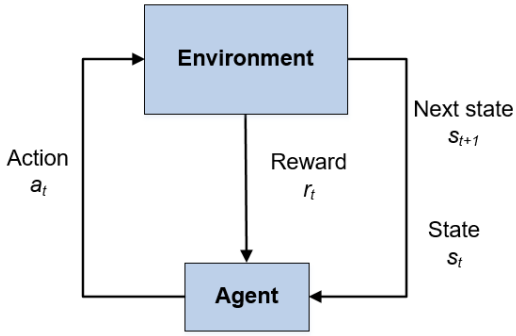
đo về mức độ di động của nút (Mobility Factor) và số lần truyền dự kiến (Expected Transmission Count) làm phần thưởng để đề xuất thuật toán định tuyến AQ-Routing dựa trên Q-Learning cho MANET. Thuật toán đề xuất đã cải thiện được thông lượng và chọn được tuyến đường ổn định hơn. Các tác giả [9] đề xuất thuật toán định tuyến sử dụng RL cho mạng MANET dựa trên 5G. Mỗi nút sử dụng RL để cập nhật cơ sở dữ liệu về tải lưu lượng và SNIR tại các nút trung gian dọc tuyến đường tới nút đích. Khi khám phá ra một con đường mới, thuật toán định tuyến tham khảo cơ sở dữ liệu này để tìm lộ trình đảm bảo QoS. Các kết quả mô phỏng chứng minh rằng thuật toán được đề xuất đạt hiệu quả về mặt mạng thông lượng, độ trễ đầu - cuối và SNIR.

Hơn nữa, các phương pháp tiếp cận dựa trên RL thể hiện khả năng thích ứng với điều kiện mạng động, đảm bảo định tuyến hiệu quả ngay cả trong MANET có tính di động cao và không thể đoán trước kịch bản. Nghiên cứu này cung cấp một cách tiếp cận về việc khai thác kỹ thuật RL để cải thiện hiệu quả và độ tin cậy của các giao thức định tuyến trong mạng MANET. Các phần tiếp theo của bài báo được bố cục như sau: Mục II trình bày phương pháp học tăng cường và ứng dụng của nó trong định tuyến MANET. Mục III là một số kết quả đánh giá bằng mô phỏng trên NS2. Cuối cùng, Mục IV là kết luận và đề xuất các hướng nghiên cứu tiếp theo.

II. ỨNG DỤNG CỦA HỌC TĂNG CƯỜNG TRONG ĐỊNH TUYẾN MANET

A. Học tăng cường

Học tăng cường [10, 1-2] là một lĩnh vực con của học máy, ở đó hệ thống học từ các hành động trước đó của nó để chọn hành động tốt hơn trong tương lai. Bản chất của RL là thử và sai, nghĩa là lặp lại các thử nghiệm và học hỏi từ mỗi thử nghiệm. Hình 2 minh họa nguyên tắc làm việc của RL trong đó một tác tử hoạt động như một người học, tương tác với môi trường để chọn một hành động sao cho phần thưởng thu được là lớn nhất.



Hình 2. Mô hình hoạt động của học tăng cường [10]

Mô hình của học tăng cường gồm có 3 thành phần chính: Tác nhân (agent), môi trường (environment) và giá trị phản hồi (reward). Quá trình học là một quá trình lặp đi lặp lại các hành động. Ban đầu tác nhân hạn chế kiến thức về môi trường quanh nó, cố gắng khám phá bằng cách quan sát trạng thái môi trường hiện tại và thực hiện hành động dựa trên quan sát này. Sau khi thực hiện mỗi hành động thì tác nhân nhảy từ trạng thái này sang trạng thái khác, đồng thời nhận được giá trị phản hồi (reward) từ hành động cũ. Dựa vào các giá trị phản hồi nhận được tác nhân có thể điều chỉnh luật chọn hành động (policy) của mình trong các bước tiếp theo. Việc điều chỉnh và tối ưu hóa luật chọn hành động dựa vào các giá trị phản hồi chính là quá trình học tăng cường.

Thông thường, trong RL môi trường được mô hình hóa như mô hình Markov (MDP - Markov Decision Process) [10]. MDP được biểu diễn bởi bộ 4 tham số (S, A, P, R). Trong đó: S là tập các trạng thái (states), A là tập các hành động (actions), P là phân bố xác suất khi chuyển đổi từ trạng thái s sang trạng thái kế tiếp s' sau khi thực hiện hành động a, và R là phần thưởng (reward) nhận được tức thì khi chuyển trạng thái từ s sang s'.

Trong phương pháp RL, tổng phần thưởng sau khi thực hiện hành động a_t (a_t được chọn theo chính sách- policy) ở trạng thái s_t là $Q(s_t, a_t)$, được xác định bởi Q-Learning [11] là một thuật toán RL. $Q(s_t, a_t)$ được gọi là giá trị Q, cập nhật theo:

$$Q(s_t, a_t) = (1-\alpha) \times Q(s_t, a_t) + \alpha \times [R(s_t, a_t) + \gamma \times max\ Q(s_{t+1}, a_{t+1})] \quad (1)$$

với α và γ là tỉ lệ học và hệ số chiết khấu, α và $\gamma \in [0,1]$, $R(s_t, a_t)$ là phần thưởng nhận được tức thì khi thực hiện hành động a_t tại trạng thái s_t, $Q(s_{t+1}, a_{t+1})$ là giá trị Q khi thực hiện hành động a_{t+1} ở trạng thái tiếp theo s_{t+1}.

B. Định tuyến MANET sử dụng học tăng cường

Trong bài toán định tuyến MANET, tác nhân học tập là các nút tương tác với môi trường chính là hệ thống mạng, hành động thực hiện là chọn nút láng giềng để chuyển tiếp gói tin tới đích. Trong học tăng cường, một tác nhân được mô hình hóa như một bộ ba bao gồm {*trạng thái, hành động, phần thưởng*}. Trong đó phần thưởng tức thì được chúng tôi chọn dựa vào nhiệm vụ gửi gói dữ liệu đến được nút đích, tổng phần thưởng được cập nhật tại mỗi nút theo thuật toán Q-Learning.

1. Trạng thái

Trạng thái: $s^i_t \in S = \{M_i \ \ i = 1, 2,\ldots, N\}$, được xác định bởi tổng số láng giềng của mỗi nút. M_i là tổng số nút láng giềng của nút i, N là tổng số nút mạng.

2. Hành động

Hành động: $a^i_t \in A = \{1, 2,\ldots, J\}$, mỗi hành động a^i_t đại diện cho việc lựa chọn nút chuyển tiếp j từ các láng giềng của i. J là số nút láng giềng của i.

Trong hình 3, tất cả các hành động có thể có của nút $i = 1$ là $a^i_t \in A = \{2, 3, 4\}$, giả sử nút $i = 1$ muốn thiết lập tuyến đến nút 6.

3. Phần thưởng

Việc học của một nút chịu ảnh hưởng chính từ phần thưởng nhận được từ nút láng giềng. Trong thiết kế của chúng tôi, với mong muốn tìm được đường đi ngắn nhất cho việc gửi dữ liệu, phần thưởng được tính dựa vào kết quả của việc gửi gói dữ liệu đến được nút đích. Nút nào trực tiếp chuyển gói dữ liệu cho nút đích sẽ nhận được phần thưởng lớn nhất.

Đặt $R(i, j)$ là phần thưởng cho hành động nút i chọn nút kế tiếp j để gửi gói dữ liệu và được tính dựa theo (2):

$$R(i,j) = \begin{cases} 100, & \text{nếu } j \text{ là nút đích} \\ -100, & \text{nếu } j \text{ chỉ có 1 láng giềng là } i \\ 0, & \text{trường hợp còn lại} \end{cases} \quad (2)$$

Theo công thức tính thưởng trên, giá trị thưởng $R(i, j)$ là lớn nhất (100) nếu j là nút đích. Trường hợp j không phải là nút đích và chỉ có duy nhất một láng giềng là i (đây là trường hợp gặp hố định tuyến) thì giá trị thưởng (-100) ở đây được chọn mang tính "phạt" nút i cho hành động đã chọn nhầm nút j làm nút kế tiếp. Cho trường hợp còn lại, j là nút trung gian nên gán phần thưởng là 0.

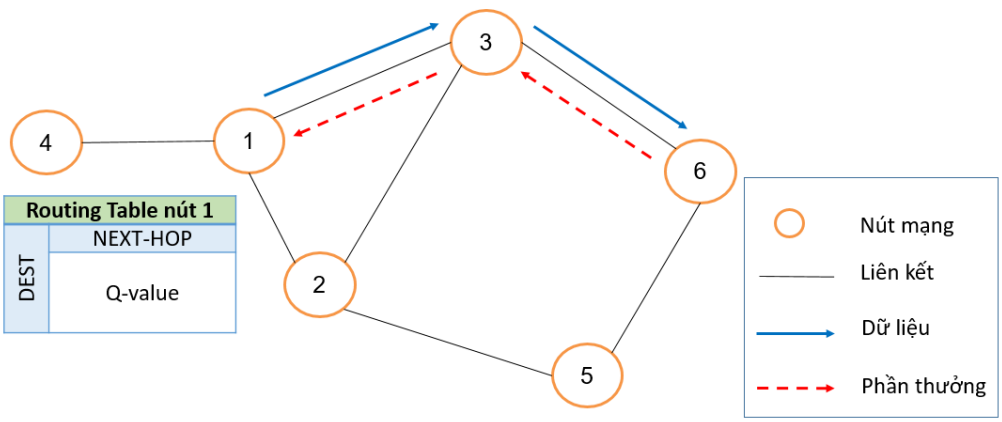
Tại nút i, tác tử học và cập nhật bảng định tuyến (Q-table) theo công thức (3):

$$Q(i,j,d) = (1-\alpha) \times Q(i,j,d) + \alpha \times [R(i,j) + \gamma \times \max_{\forall k \in L_j} Q(j,k,d)] \quad (3)$$

Với $Q(i,j,d)$ là tổng phần thưởng từ nút i đến nút đích d và j là nút kế tiếp ($i \rightarrow j \rightarrow \ldots \rightarrow d$). $R(i, j)$ xác định theo công thức (2). L_j là tập các nút láng giềng của nút j. α và γ là tỉ lệ học và hệ số chiết khấu, α và $\gamma \in [0,1]$.

Dựa vào giá trị Q ($Q(i,j,d)$) này giúp cho các nút lựa chọn nút kế tiếp để gửi dữ liệu, Q càng lớn thì đường đi đến nút đích càng gần.

Trong quá trình lựa chọn nút kế tiếp j, có hai cách là khai thác và thăm dò. Khai thác là chọn nút kế tiếp j với giá trị Q là lớn nhất. Thăm dò là chọn j ngẫu nhiên trong số các láng giềng của i. Bài báo này sử dụng phương pháp ε-*greedy* để thực hiện hành động chọn nút kế tiếp, trong đó tác nhân thực hiện khai thác với xác suất nhỏ hơn ε và thăm dò với xác suất 1 - ε. Tỉ lệ học α, hệ số chiết khấu γ và ε được đặt cố định tương ứng là 0,8, 0,8 và 0,9.

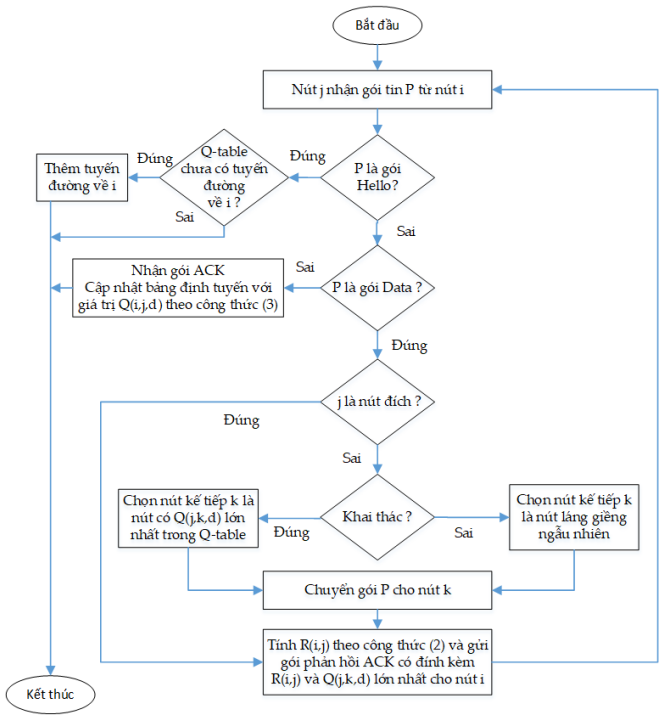


Hình 3. Kịch bản định tuyến dựa trên học tăng cường

Hình 3 minh họa mô hình RL được tích hợp vào quá trình định tuyến tại các nút mạng. Giả sử nút 1 muốn gửi dữ liệu đến nút đích 6. Nút 1 có thể chọn nút láng giềng 3 để gửi gói dữ liệu ($a^i_t = 3$), và nhận phần thưởng từ nút 3 đó là ước tính chi phí của tuyến đường từ nút 3 đến nút đích 6 (cụ thể là tuyến đường 3 → 6). Sau đó, nút 1 cập nhật giá trị Q của nó cho hành động $a^i_t = 3$ sử dụng công thức (3), cụ thể là $Q(1, 3, 6)$. Tương tự như vậy, khi nút 1 gửi các gói dữ liệu của nó đến nút $a^i_t = 2$ nó nhận được phần thưởng là ước tính chi phí của tuyến đường từ nút 2 đến nút đích 6 có thể là tuyến đường (2 → 3 → 6) hoặc (2 → 5→ 6) và cập nhật giá trị $Q(1, 2, 6)$. Lưu ý rằng, nút 2 chọn tuyến đường (2 → 3 →

6) hoặc (2 → 5→ 6) phụ thuộc vào giá trị $Q(2, 3, 6)$ và $Q(2, 5, 6)$ tại nút 2. Nếu chọn giá trị Q lớn nhất là hành động khai thác.

4. Lưu đồ thuật toán xử lý gói tin và học từ gói ACK tại nút *j*



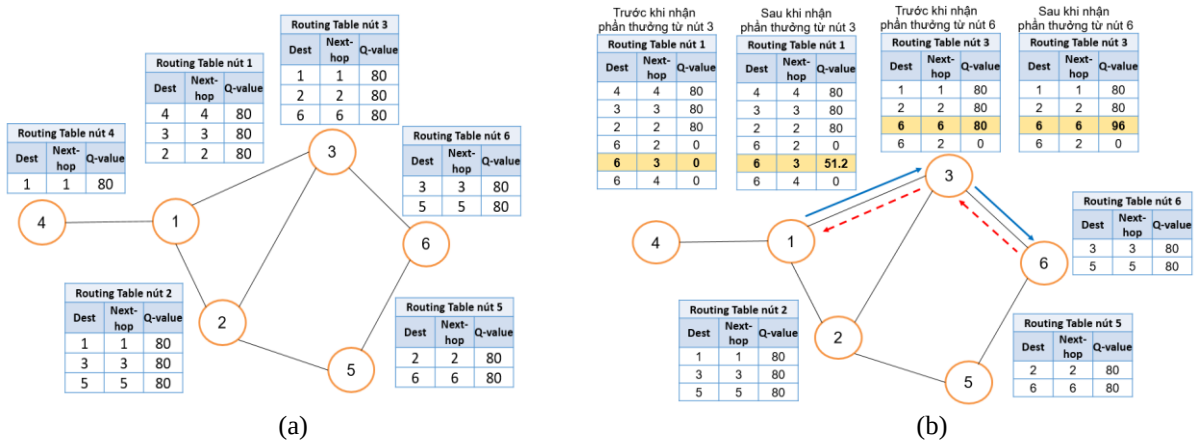
Hình 4. Sơ đồ thuật toán xử lý gói tin và học từ gói phản hồi ACK tại nút *j*

Khi nút *j* nhận được gói tin P từ nút *i* (P là 1 trong 3 loại gói: Hello, ACK, Data), quá trình xử lý như sau:

- Nếu P là gói Hello, kiểm tra trong bảng định tuyến xem đã có tuyến đường về nút *i* với nút kế tiếp là *i* chưa. Nếu chưa có thì thêm vào với giá trị *Q* được tính theo công thức (3).

- Nếu P là gói ACK, sẽ trích xuất thông tin $R(i,j)$ và giá trị Q lớn nhất, cập nhật giá trị $Q(i,j,d)$ tương ứng theo công thức (3).

- Nếu P là gói Data, kiểm tra xem *j* có phải là nút đích không. Nếu *j* là nút đích, tính $R(i,j)$ theo công thức (2) và gửi gói phản hồi ACK có đính kèm $R(i,j)$ và giá trị Q lớn nhất cho nút *i*. Nếu *j* là nút trung gian, so sánh nếu xác suất nhỏ hơn ε (0,9) thì khai thác bằng cách chọn nút kế tiếp *k* là nút có giá trị Q lớn nhất trong bảng định tuyến để gửi gói P. Ngược lại là khám phá bằng cách chọn ngẫu nhiên nút kế tiếp *k* là 1 trong các láng giềng của *j* và gửi gói P. Sau khi gửi gói P, nút *j* tính $R(i,j)$ theo công thức (2) và gửi gói phản hồi ACK có đính kèm $R(i,j)$ và $Q(j,k,d)$ lớn nhất cho nút *i*.



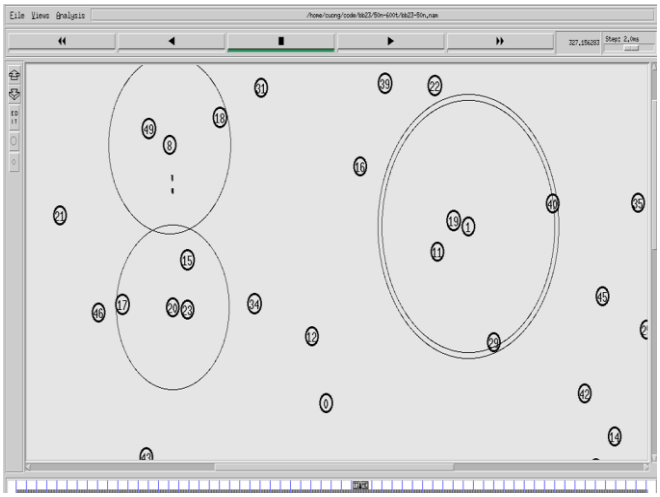
Hình 5. Một ví dụ định tuyến dựa trên học tăng cường

Để minh họa cho quá trình định tuyến, xét ví dụ như Hình 5, trong đó nút 1 muốn gửi dữ liệu cho nút 6. Trước thời điểm gửi dữ liệu, các nút gửi các gói Hello để xác định các nút láng giềng đồng thời thêm vào bảng định tuyến các tuyến đường đến các láng giềng của nó như Hình 5 (a). Hình 5 (b) thể hiện quá trình nút 1 gửi gói dữ liệu cho nút 3 và nút 3 chuyển tiếp cho nút 6, xét thời điểm nút 1 muốn gửi dữ liệu cho nút 6, lúc này bảng định tuyến nút chưa có tuyến đường đến nút 6 nên nó khởi tạo các tuyến đường tới nút 6 với nút kế tiếp là các láng giềng của nó (nút 2, 3, 4) và giá

trị Q = 0 [là các dòng (6, 2, 0); (6, 3, 0); (6, 4,0) trong bảng định tuyến]. Tại nút 1 có 3 tuyến đường đều có giá trị Q bằng 0 nên nó chọn ngẫu nhiên nút kế tiếp để gửi dữ liệu (giả sử chọn nút 3), tuyến đường lúc này là (1 → 3). Khi nút 3 nhận được gói dữ liệu, nó thêm vào bảng định tuyến 1 tuyến đường tới nút 6 qua nút kế tiếp là láng giềng 2 với giá trị Q bằng 0 [là dòng (6, 2, 0)]. Đồng thời nút 3 gửi gói phản hồi phần thưởng cho nút 1 với giá trị Q lớn nhất là 80 và R bằng 0 (nút 3 chưa là nút đích). Nút 1 nhận gói phản hồi từ nút 3, cập nhật tuyến đường [dòng (6, 3, 0) thành (6, 3, 51.2)] theo công thức (3). Nút 3 xét bảng định tuyến và chọn nút kế tiếp là 6 (khai thác với Q lớn nhất) để gửi dữ liệu, tuyến đường lúc này là (1 → 3 → 6). Nút đích 6 nhận dữ liệu và gửi phản hồi cho nút 3 giá trị Q lớn nhất là 0 và R bằng 100 (nút 6 là nút đích), nút 3 nhận gói phản hồi và cập nhật tuyến đường [dòng (6, 6, 80) thành (6, 6, 96)].

III. ĐÁNH GIÁ KẾT QUẢ BẰNG MÔ PHỎNG

Để đánh giá hiệu quả của giao thức sử dụng mô hình RL được trình bày ở trên, chúng tôi tiến hành cài đặt, mô phỏng trên hệ mô phỏng NS2 phiên bản 2.35 [12], giao thức được đặt tên là RLMR (Reinforcement Learning-based MANET Routing). RLMR được so sánh với giao thức cơ bản của MANET là DSDV về tỉ lệ gửi gói dữ liệu thành công (PDR: Packet Delivery Ratio), thông lượng mạng (Throughput) và thời gian trễ đầu cuối (EED: End-to-End Delay). Với các thông số mô phỏng được trình bày như trong Bảng 1. Hình 6 thể hiện giao diện mô phỏng với kịch bản có 50 nút mạng. Các kịch bản mô phỏng được chạy với số lần là 10 và các kết quả số liệu trong bài báo này là giá trị trung bình của 10 lần chạy. Tỉ lệ học tập α, hệ số chiết khấu γ được đặt là 0,8 và ε là 0,9.



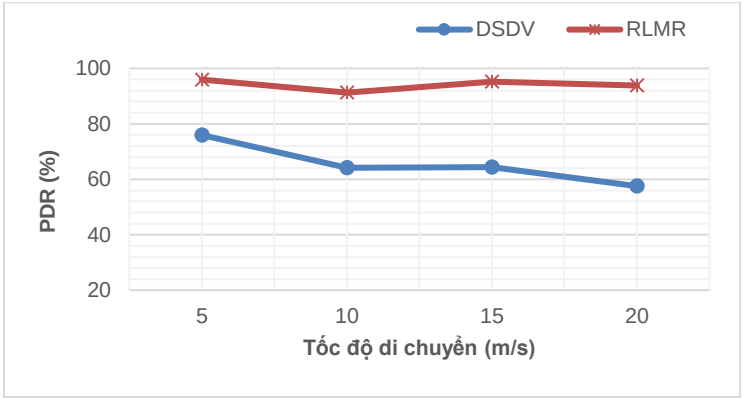
Hình 6. Giao diện mô phỏng trên NS2

Bảng 1. Thông số thiết lập mô phỏng

Thông số	**Giá trị**
Giao thức định tuyến	RLMR, DSDV
Khu vực địa lý	1000 m x 1000 m
MAC protocol	802.11
Số nút mạng	30, 40, 50
Tốc độ di chuyển	0-20 m/s
Bán kính phát sóng	250 m
Mô hình di động	Random Waypoint
Thời gian mô phỏng	600 giây
Giao thức vận chuyển	UDP
Loại nguồn phát	CBR
Kích thước gói dữ liệu	512 byte
Tốc độ phát	4 gói/giây
Số lần chạy 1 kịch bản	10

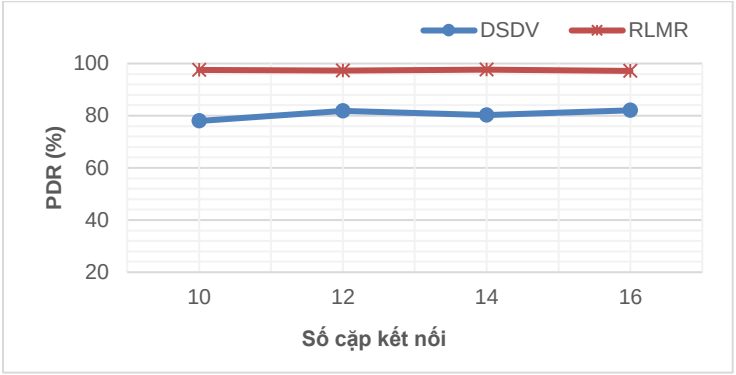
Sau khi thực hiện các kịch bản mô phỏng, số liệu được thu thập và xử lý cho kết quả như sau:

Tỉ lệ gửi gói dữ liệu thành công (PDR)

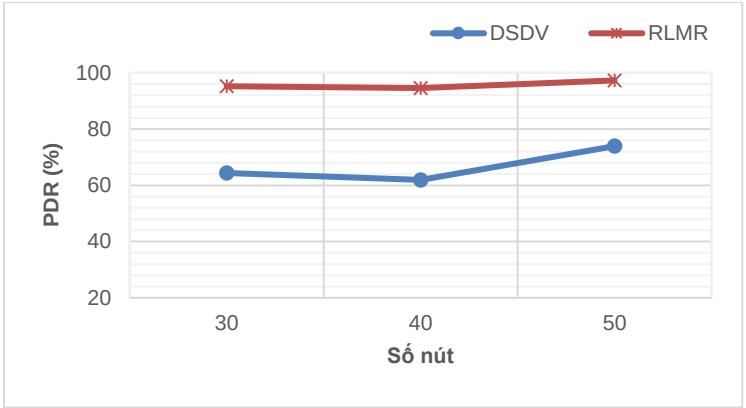


Hình 7. Tỉ lệ gửi gói dữ liệu thành công với số nút 30

Tỉ lệ gửi gói dữ liệu thành công được thể hiện trong Hình 7, 8 và Hình 9. Kết quả cho thấy, giao thức RLMR có tỉ lệ cao hơn so với giao thức DSDV. Xét trường hợp với 30 nút mạng di chuyển với các tốc độ trung bình là 5, 10, 15, 20 m/s (Hình 7), tỉ lệ gửi gói dữ liệu thành công trunh bình của RLMR là 94,07% so với DSDV là 65,51%. Trong trường hợp số nút mạng là 40, với số cặp kết nối đầu - cuối thay đổi từ 10, 12, 14, 16 (Hình 8), RLMR vẫn luôn có tỉ lệ PDR cao hơn trung bình 16% so với DSDV. Khi tăng số nút lên từ 30 lên 40 và lên 50 nút với tốc độ di chuyển trung bình của các nút là 20 m/s, tỉ lệ gửi gói của RLMR vẫn được mức ổn định và luôn cao hơn so với DSDV được thể hiện trong Hình 9.

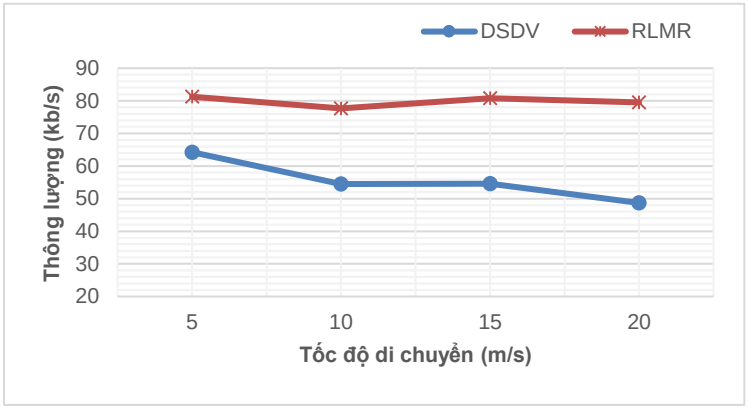


Hình 8. Tỉ lệ gửi gói dữ liệu thành công với số cặp kết nối thay đổi

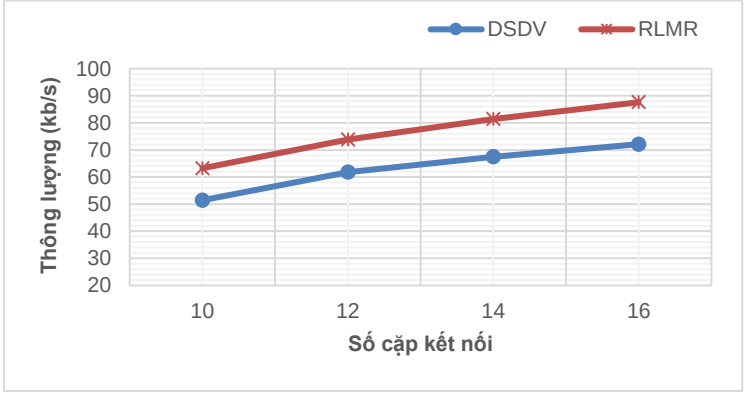


Hình 9. Tỉ lệ gửi gói dữ liệu thành công với tốc độ di chuyển 15m/s

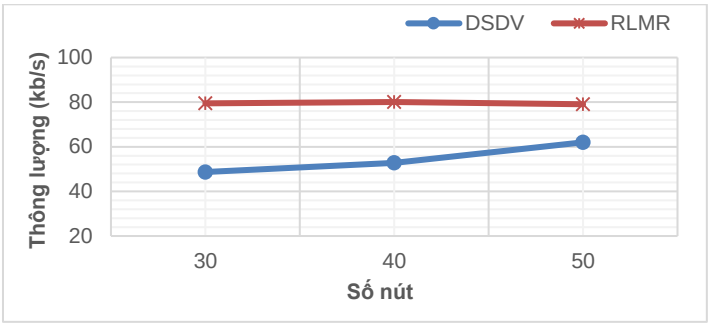
Thông lượng



Hình 10. Thông lượng với số nút 30



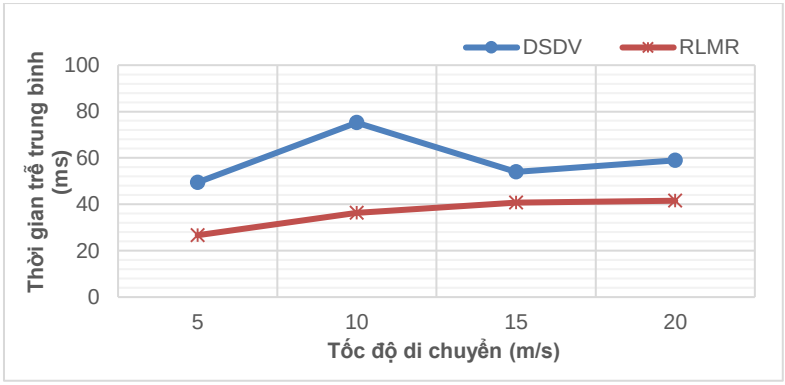
Hình 11. Thông lượng với số cặp kết nối thay đổi



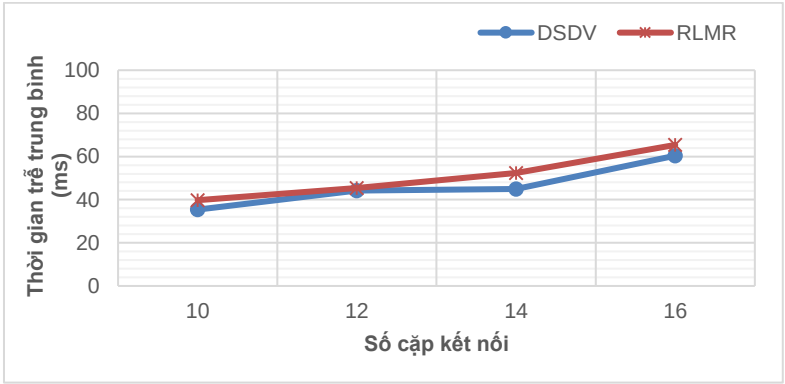
Hình 12. Thông lượng với tốc độ di chuyển 20 m/s

Hình 10 thể hiện kết quả thông lượng trung bình của các giao thức RLMR và DSDV với kịch bản mô phỏng có 30 nút mạng di chuyển với các tốc độ khác nhau (5, 10, 15, 20 m/s). Giao thức RLMR đạt mức trung bình 79,79 kb/s cao hơn hẳn so với DSDV là 55,51 kb/s. Hình 11 cho thấy, khi số cặp kết nối đầu - cuối tăng lên (10, 12, 14, 16) với 40 nút mạng, thông lượng trung bình cả hai giao thức đều tăng nhưng RLMR vẫn luôn cao hơn DSDV. Khi số nút và tốc độ di chuyển tăng lên thông lượng trung bình của RLMR vẫn đạt được mức ổn định và cao hơn trung bình 25 kb/s so với DSDV như trong Hình 12.

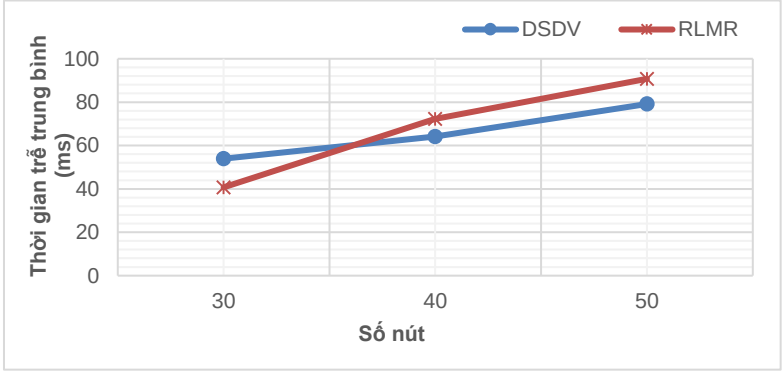
Thời gian trễ trung bình (EED)



Hình 13. Thời gian trễ trung bình với số nút 30



Hình 14. Thời gian trễ trung bình với số cặp kết nối thay đổi



Hình 15. Thời gian trễ trung bình với tốc độ di chuyển 15 m/s

Cuối cùng, chúng tôi phân tích thời gian trễ trung bình đầu cuối của 2 giao thức. Trong Hình 13 cho thấy thời gian trễ trung bình của giao thức RLMR là tốt hơn với mức 36,31 ms so với DSDV là 69,39 ms. Ở Hình 14, khi tăng số nút mạng lên 40 với số cặp kết nối tăng dần từ 10, 12, 14, 16 thì EED của RLMR lại cao hơn so với DSDV, chênh lệch trung bình 4,48 ms. Trong trường hợp, tăng số nút mạng từ 30 lên 40, 50 thì EED của cả 2 giao thức đều tăng lên, trong đó RLMR tăng lên cao hơn so với DSDV (Hình 15). Nguyên nhân làm cho EED của RLMR tăng lên là do khi số nút tăng, số láng giềng của mỗi nút cũng tăng lên, giai đoạn đầu khi các nút bắt đầu quá trình học (lúc này giá trị Q của các nút láng giềng đều bằng 0) nút chọn ngẫu nhiên một nút kế tiếp trong số các láng giềng đó nên tốn nhiều thời gian hơn để chọn được nút kế tiếp tốt nhất. Mặt khác, trong học tăng cường, giao thức thực hiện cả khai thác và khám phá, khi thực hiện khám phá với số lượng nút tăng làm gia tăng thời gian của quá trình khám phá.

IV. KẾT LUẬN

Qua việc nghiên cứu học tăng cường và áp dụng mô hình học tăng cường cho định tuyến MANET, bài báo đưa ra cách tính phần thưởng dựa vào kết quả của nhiệm vụ gửi gói dữ liệu đến được nút đích. Dựa vào đó, chúng tôi đã cài đặt, đánh giá được hiệu năng của giao thức RLMR so với giao thức DSDV. Kết quả mô phỏng cho thấy rằng giao thức RLMR đạt hiệu suất cao hơn về mặt tỉ lệ gửi tin gói thành công và thông lượng mạng. Đối với thời gian trễ trung bình đầu cuối trong kịch bản tăng số nút mạng, RLMR chưa được cải thiện, chúng tôi sẽ tìm hướng cải thiện trong các nghiên cứu tiếp theo.

Trong thời gian tới chúng tôi tiếp tục nghiên cứu ứng dụng học tăng cường cho định tuyến MANET, với các tham số định tuyến phù hợp để làm phần thưởng cho việc học, xem xét các yếu tố ảnh hưởng để thay đổi linh hoạt hệ số học và hệ số chiết khấu.

TÀI LIỆU THAM KHẢO

[1] Hoebeke J., Moerman I., Dhoedt B., Demeester P., "An Overview of Mobile Ad Hoc Networks: Applications and Challenges," *Journal of the Communications Network*, vol. 3(3), pp. 60-66. 2004.

[2] S.K. Sarkar, T.G. Basavaraju, C. Puttamadappa, *Ad Hoc Mobile Wireless Networks: Principles, Protocols, and Applications*. Taylor & Francis Group, LLC, 2008.

[3] Forster A, "Machine learning techniques applied to wireless ad-hoc networks: guide and survey," in *Proceedings of ISSNIP 3rd international conference intelligent sensors, sensor Networks and information*, pp. 365-370, 2007.

[4] R. Mili, S. Chikhi, *Reinforcement learning based routing protocols analysis for mobile ad-hoc networks.* in E. Renault, P. Mühlethaler, S. Boumerdassi (Eds.), Machine Learning for Networking, Springer, pp. 247-256, 2019.

[5] S. Chettibi, S. Chikhi, "An adaptive energy-aware routing protocol for MANETs using the SARSA reinforcement learning algorithm," in *2012 IEEE Conference on Evolving and Adaptive Intelligent Systems*, pp. 84-89, 2012.

[6] M. Yin, J. Chen, X. Duan, B. Jiao, Y. Lei, "QEBR: Q-Learning based routing protocol for energy balance in wireless mesh networks," in *2018 IEEE 4th International Conference On Computer And Communications, ICCC*, pp. 280-284, 2018.

[7] T.V.T. Duong, L.H. Binh, and V.M. Ngo, "Reinforcement learning for QoS-guaranteed intelligent routing in Wireless Mesh Networks with heavy traffic load," *ICT Express*, 8(1), pp. 18-24, 2022.

[8] Serhani, Abdellatif, Najib Naja, and Abdellah Jamali, "AQ-Routing: mobility-, stability-aware adaptive routing protocol for data routing in MANET-IoT systems," *Cluster Computing*, 23.1, pp. 13-27, 2020.

[9] L.H. Binh and T.V.T. Duong, "An improved method of AODV routing protocol using reinforcement learning for ensuring QoS in 5G-based mobile ad-hoc networks," *ICT Express*, 2023. https://doi.org/10.1016/j.icte.2023.07.002.

[10] Richard S. Sutton and Andrew G. Barto., *Reinforcement Learning: An Introduction.* The MIT Press Cambridge, Massachusetts London, England, 2018.

[11] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, no. 3-4, pp. 279-292, 1992.

[12] DARPA, "The Network simulator ns-allinone," 2.35. [Online]. Available: http://www.isi.edu/nsnam/ns

RLMR: A METHOD OF APPLYING Q-LEARNING FOR ROUTING IN MOBILE ADHOC NETWORKS

Nguyen Quoc Cuong, Mai Cuong Tho, Le Huu Binh, Vo Thanh Tu

ABSTRACT: *Currently, researching reinforcement learning for routing protocols has garnered significant attention. In MANET, the high mobility of node to dynamic and unstable link structures, posing a challenge that necessitates routing protocols to adapt swiftly. Reinforcement learning has been demonstrated to address this routing challenge by enabling network nodes to observe and gather information from their local operational environment, learn, and make routing decisions effectively. This article focuses on applying a reinforcement learning model for routing in MANETs to enhance network performance. Simulation results using NS2 indicate that the routing protocol incorporating reinforcement learning has improved performance in terms of data packet delivery rates and network throughput.*