

MÔ HÌNH MẠNG GIAO THÔNG THÔNG MINH BA TẦNG VÀ GIẢI PHÁP GIẢM THIỂU ĐỘ TRỄ TRUYỀN TIN ỨNG DỤNG PHƯƠNG PHÁP HỌC TĂNG CƯỜNG

Nguyễn Năng Hưng, Nguyễn Trung Thành, Phạm Minh Khiêm, Đinh Văn Hiếu,
Nguyễn Đình Tuấn Anh, Nguyễn Thanh Hùng, Nguyễn Phi Lê
Trường Công nghệ thông tin và Truyền thông, đại học Bách Khoa Hà Nội

Tóm tắt—Ứng dụng công nghệ điện toán đám mây di động (MCC) và điện toán biên di động (MEC) để giảm tải tác vụ tính toán là hướng tiếp cận đầy hứa hẹn cho phép các ứng dụng yêu cầu độ trễ thấp có thể thực thi trên các phương tiện giao thông thông minh. Hầu hết các nghiên cứu về giảm tải hiện nay tập trung vào sử dụng một mô hình đơn nhất là MEC hoặc MCC. Trong báo cáo này, chúng tôi nghiên cứu việc sử dụng kết hợp các công nghệ trong một mạng giao thông thông minh ba tầng V2X, trong đó phương tiện giao thông có thể giảm tải các tác vụ tính toán bằng cả MEC và MCC. Cụ thể, chúng tôi xem xét các xác suất tối ưu để định tuyến truyền tải trên ba đường truyền tin bao gồm: V2I, V2N và I2N. Nghiên cứu của chúng tôi có các đóng góp chính: Thứ nhất, chúng tôi đề xuất mô hình toán học tính toán chính xác độ trễ xử lý của các tác vụ, và công thức tìm lời giải tối ưu để đạt được độ trễ tối thiểu. Thứ hai, chúng tôi đề xuất phương pháp tối ưu dựa trên giải thuật di truyền (GA). Thứ ba, nhận thấy yếu điểm của GA, chúng tôi đề xuất một giải thuật ứng dụng kỹ thuật học tăng cường: multi-agent multi-armed bandits (MAB) với cải tiến cơ chế khám phá dựa trên hàm Sigmoid (SEM) để tối thiểu hóa độ trễ. Kết quả thí nghiệm cho thấy rằng thuật toán GA chúng tôi đề xuất trong mô hình ba tầng có thể rút ngắn độ trễ lên đến **99,9%** so với các mô hình hai tầng hiện có, cơ chế khám phá cải tiến SEM chúng tôi đề xuất có độ trễ các tác vụ trung bình giảm **35%** so với MAB cơ bản, thí nghiệm cũng cho thấy thuật toán MAB cải tiến rút ngắn độ trễ với trung bình **18.5%** và **56.9%** trong trường hợp tốt nhất so với thuật toán GA.

Từ khóa—Học tăng cường, MEC, V2X.

I. GIỚI THIỆU

Sự phát triển mạnh mẽ của mạng không dây trong những năm gần đây đã tạo nên móng đầy hứa hẹn cho các ứng dụng vào hệ thống tính toán cường độ lớn như hệ thống

giao thông thông minh và xe tự hành. Những ứng dụng, hệ thống như vậy đòi hỏi một khối lượng tài nguyên điện toán khổng lồ để đáp ứng được các tác vụ ra quyết định thời gian thực hoặc xử lý một lượng lớn dữ liệu cảm biến. Những yêu cầu đó là một thách thức lớn đối với các phương tiện giao thông chỉ có khả năng tính toán giới hạn. Một giải pháp cho vấn đề này là đặt các đơn vị điện toán biên (Roadside Units - RSU) ở ven đường, nơi có khoảng cách tương đối nhỏ tới các phương tiện giao thông. Các đơn vị RSU này có khả năng tính toán tương đối lớn, cung cấp dịch vụ tính toán có độ trễ thấp và độ tin cậy truyền tin cao. Kết quả là, các phương tiện bị giới hạn tính toán có thể chuyển tác vụ của mình tới các đơn vị RSU để có thể đáp ứng nhu cầu điện toán và thời gian xử lý.

Tuy nhiên, các RSU cũng chỉ có một khả năng tính toán cùng với phạm vi truyền tin có giới hạn nhất định. Vì vậy, một mạng lưới các RSU không thể đáp ứng được tốc độ xử lý trong trường hợp phải tính toán một lượng lớn các tác vụ từ các phương tiện giao thông, hệ quả là độ trễ các gói tin tăng lên đáng kể. Giải pháp cho vấn đề này là liên kết các RSU với một trạm xử lý trung tâm, ví dụ như một gNodeB (gNB), có khả năng cung cấp dịch vụ tính toán tốc độ cao, khối lượng xử lý lớn và phạm vi bao quát, truyền tin không giới hạn. Trong trường hợp các RSU quá tải xử lý các tác vụ, có khả năng gây trì trệ xử lý cao, thì các RSU có thể chuyển một phần các tác vụ của mình tới trung tâm xử lý gNB, vừa giúp giảm áp lực đường truyền tầng dưới, vừa giảm khối lượng tính toán cho các RSU. Trong mô hình ba tầng như vậy (phương tiện giao thông, đơn vị điện toán biên, trung tâm xử lý), một phương tiện giao thông cũng có thể chuyển tác vụ của mình trực tiếp lên gNB để xử lý với khả năng điện toán vô cùng lớn. Một hạn chế của việc chuyển tác vụ lên gNB là độ trễ truyền tin lớn khi các phương tiện cách xa trung tâm xử lý gNB.

Lấy cảm hứng từ bài toán này, nhóm nghiên cứu đề xuất mô hình ba tầng tương ứng để giải quyết vấn đề tối ưu cả độ trễ truyền tin và vấn đề phân bổ tài nguyên tính toán. Cụ thể trong bài nghiên cứu này, nhóm sử dụng mô hình gồm ba đường truyền tín hiệu: V2I, I2N và V2N. Với V2I ký hiệu cho đường truyền từ các phương tiện giao thông với các đơn vị điện toán biên RSU, I2N ký hiệu cho đường truyền từ các đơn vị điện toán biên RSU tới trung tâm xử

Tác giả liên hệ: Nguyễn Thanh Hùng, Nguyễn Phi Lê,
Email: {hungnt, lenp}@soict.hust.edu.vn
Đến tòa soạn: 1/6/2021, chỉnh sửa: 15/8/2021, chấp nhận đăng:
31/8/2021.

lý gNB, và V2N ký hiệu cho đường truyền trực tiếp từ các phương tiện giao thông tới trung tâm xử lý gNB. Sau khi RSU hoặc gNB xử lý xong một tác vụ, kết quả được gửi trả về các phương tiện giao thông. Nghiên cứu được thực hiện trong bài hướng đến tối thiểu hóa độ trễ trung bình của tất cả các gói tin luân chuyển trong mạng, nghiên cứu của chúng tôi có những đóng góp như sau:

- 1) Xây dựng mô hình toán học mô phỏng lại quá trình luân chuyển các tác vụ trong mô hình mạng ba tầng. Thiết kế biểu thức tính toán chính xác độ trễ truyền tin dưới dạng hàm số với biến là các biến xác suất truyền tin.
- 2) Đề xuất phương pháp sử dụng giải thuật di truyền (Genetic Algorithm - GA) để xấp xỉ giá trị cho các biến xác suất để tối ưu hàm độ trễ đã xây dựng.
- 3) Đánh giá ưu điểm, nhược điểm của phương pháp sử dụng GA để giải quyết mục tiêu tối ưu của bài toán. Đề xuất phương pháp ứng dụng học tăng cường (Reinforcement Learning - RL) và một cải tiến của phương pháp này (Sigmoid Exploration Mechanism - SEM) để khắc phục nhược điểm của phương pháp GA.
- 4) Đánh giá kết quả của GA và SEM dựa trên kết quả thí nghiệm giả lập, so sánh ưu điểm của mô hình ba tầng so với mô hình hai tầng (phương tiện - RSU; phương tiện - gNB).

II. CÁC NGHIÊN CỨU LIÊN QUAN

Trong những năm gần đây, có rất nhiều các công trình nghiên cứu về mạng lưới phương tiện giao thông dựa trên kiến trúc MEC. Trong nghiên cứu của Feng và cộng sự [1] đã xem xét đến các vấn đề truyền tin và tối ưu hóa phân bổ tài nguyên. Với mục tiêu giảm thiểu mức độ tiêu thụ điện năng cho các phương tiện và máy chủ MEC, nhóm tác giả đã chia vấn đề chính thành hai vấn đề nhỏ hơn. Vấn đề đầu tiên là phương thức tối ưu dung lượng truyền tin có độ trễ thấp tới các phương tiện. Vấn đề thứ hai là quyết định phương thức truyền tin giữa các phương tiện, máy chủ MEC và giữa các máy chủ MEC với nhau. Tương tự như vậy, các tác giả trong [2] đã tận dụng mối quan hệ giữa các máy chủ MEC với nhau để đạt được tối ưu khi chia sẻ nguồn tài nguyên. Cụ thể, nhóm tác giả đã đưa ra khái niệm về kiến trúc MEC hai tầng, sau đó tận dụng mô hình Deep Q-learning để đưa ra quyết định truyền tin nhằm tối ưu hóa tài nguyên.

Liên quan đến V2V và V2I, các công trình nghiên cứu trong [3], [4], [5] đã đưa ra hướng giải quyết cho quyết định truyền tin của việc thực hiện chung các tác vụ giữa các thiết bị trung gian và máy chủ MEC. Cả hai công trình [3], [5] đã nghiên cứu cách quyết định xem các tác vụ sẽ được thực hiện xử lý trực tiếp hay chuyển tải cho các thiết bị trung gian khác hoặc máy chủ MEC được liên kết. Tuy nhiên, [3] tập trung chủ yếu vào vấn đề giảm thiểu chi phí truyền tin, trong khi [5] có mục tiêu giảm thiểu tổng mức tiêu thụ năng lượng trung bình. Trong [3], nhóm tác giả đã đưa ra công thức toán học cho bài toán dưới dạng bài toán đường đi ngắn nhất có giới hạn trên đồ thị có hướng không chứa chu trình. Sau đó, họ đã khai thác thuật toán chi phí tổng hợp (Lagrangian Relaxation-based Aggregated Cost - LARAC)

để xác định giải pháp tối ưu dưới ràng buộc thời gian thực hiện tác vụ.

Trong [5] các tác giả đã tận dụng thuật toán Lyapunov để đơn giản hóa mục tiêu nghiên cứu và sử dụng giải thuật tham lam để xác định giải pháp tối ưu. Trong nghiên cứu [4] có mục tiêu giảm thiểu độ trễ khi thực hiện truyền tin giữa các phương tiện, mỗi phương tiện luôn phải duy trì thông tin của các phương tiện bên cạnh. Khi có tác vụ mới, sẽ thực hiện tính toán độ trễ truyền tin tới các phương tiện bên cạnh. Sau đó, phương tiện bên cạnh có độ trễ tối thiểu nhất sẽ được lựa chọn để truyền tin. Mặc dù có nhiều nghiên cứu đã được thực hiện, nhưng hầu hết những nghiên cứu này chỉ dựa vào các máy chủ MEC để thực hiện truyền tin.

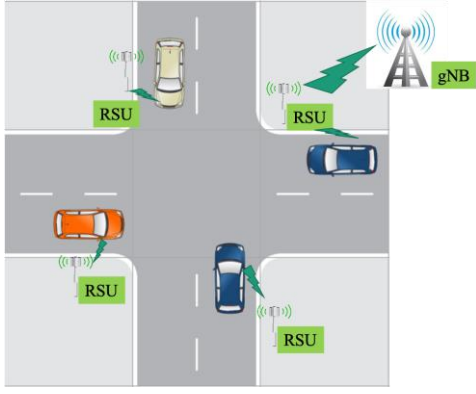
Gần đây, Zhao và cộng sự [6] đã cố gắng sử dụng đồng thời MEC và MCC để truyền tin thông qua hai con đường: V2I và I2N. Cụ thể, các phương tiện có thể tải các tác vụ tính toán lên máy chủ MEC hoặc máy chủ đám mây thông qua các RSU. Mục tiêu nhằm tối đa hóa tiện ích hệ thống bằng cách tối ưu hóa các chiến lược truyền tin và phân bổ tài nguyên. Khác với các công trình trước đây, bài báo của chúng tôi nghiên cứu mô hình ba tầng, cung cấp ba con đường truyền tin là: V2I, V2N và I2N. Mục tiêu của chúng tôi là xác định xác suất truyền tin để giảm thiểu độ trễ thực thi các tác vụ sinh ra trong mạng.

III. MÔ HÌNH LÝ THUYẾT

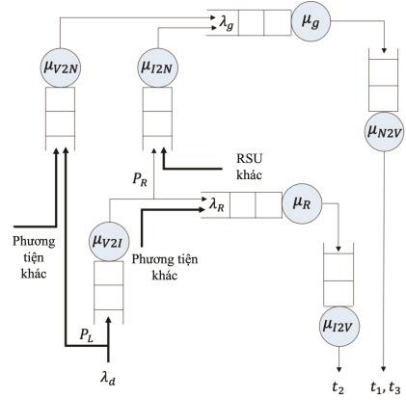
A. Mô hình mạng

1) *Tổng quan hệ thống*: Trong phần này, chúng tôi mô tả tổng quan hệ thống mạng giao thông thông minh ba tầng V2X. Hình minh họa 1(a), Ta xem xét một khu vực giao thông đô thị, có N đơn vị điện toán biên (RSU) được đặt dọc hai bên đường. Các phương tiện giao thông thông minh liên tục yêu cầu các tác vụ xử lý tính toán. Các tác vụ này có thể được chuyển lên một RSU hoặc gNB. Ngoài ra, một RSU có thể tự xử lý hoặc chuyển tiếp tác vụ này lên gNB. Sau khi xử lý xong tác vụ, kết quả được RSU hoặc gNB gửi trả trực tiếp về phương tiện giao thông ban đầu. Như vậy, một tác vụ (gói tin) có thể được phân thành ba loại như sau: (1) tác vụ phương tiện yêu cầu gNB xử lý trực tiếp, (2) tác vụ phương tiện gửi lên RSU, sau đó RSU chuyển tiếp lên gNB và xử lý tại gNB, và (3) tác vụ được phương tiện chuyển lên RSU và xử lý tại RSU. Mục tiêu của chúng tôi là tìm được giá trị tối ưu cho các xác suất truyền tin để tối thiểu độ trễ xử lý của cả ba loại tác vụ trên. Độ trễ xử lý của một tác vụ bao gồm hai phần: độ trễ đường truyền và độ trễ tính toán. Độ trễ đường truyền là tổng thời gian tác vụ di truyền trên đường truyền. Độ trễ tính toán là độ trễ từ khi đơn vị tính toán (ở đây là RSU hoặc gNB) nhận được tác vụ được yêu cầu xử lý đến khi xử lý xong tác vụ đó.

Các ký hiệu và giả định được sử dụng trong báo cáo nghiên cứu này như sau: Ký hiệu P_L và P_R là các xác suất truyền tin, trong đó P_L là xác suất phương tiện gửi tác vụ trực tiếp lên gNB, P_R là xác suất RSU chuyển tiếp tác vụ mà nó được phương tiện yêu cầu lên gNB. Chúng tôi giả định các phương tiện giao thông đi vào và rời khỏi vùng phủ sóng của các RSU tuân theo quá trình Poisson với tỉ lệ đến là λ_v và tỉ lệ đi là μ_v . Các tác vụ sinh ra ở mỗi phương tiện giao thông tuân theo phân phối Poisson với kì vọng là λ_d . Chúng tôi giả định rằng thời gian xử lý tác vụ tại RSU



(a) Mô hình mạng



(b) Mô hình hàng đợi mô phỏng luồng gói tin

Hình 1: Tổng quát mô hình mạng

và gNB tuân theo phân phối mũ với khả năng phục vụ của RSU và gNB, kí hiệu lần lượt là μ_R và μ_g . Khi các tác vụ được tải lên các RSU hoặc gNB, chúng được truyền đi thông qua các liên kết đường truyền với thời gian tuân theo quá trình Poisson, chúng tôi cũng giả sử rằng độ trễ của đường truyền liên kết tuân theo phân phối mũ. Bằng thông truyền dữ liệu từ xe lên RSU là μ_{V2I} và ngược lại từ RSU về xe là μ_{I2V} . Tương tự μ_{V2N} và μ_{N2V} tương ứng là băng thông từ xe lên gNB và ngược lại, băng thông giữa RSU-gNB là μ_{I2N} và μ_{N2I} . Cuối cùng, ta sử dụng t_1, t_2, t_3 tương ứng là thời gian trễ trung bình của các đường truyền loại (1), (2), (3) đã nêu trên.

2) *Mô hình độ trễ truyền tin*: Ở phần này, chúng ta sẽ tính toán công thức độ trễ truyền tin của ba loại đường truyền. Trước hết ta có, số lượng xe liên kết với một RSU được tính bởi công thức $E[V] = \frac{\rho_v}{1-\rho_v}$, trong đó $\rho_v = \frac{\lambda_v}{\mu_v}$. Xác suất gửi tin trực tiếp từ xe lên gNB là P_L , nên xác suất gửi tin từ xe đến RSU là $1 - P_L$. Do vậy, có thể coi số gói tin gửi lên RSU tuân theo phân phối Poisson với tham số tỉ lệ là $\lambda_{V2I} = \frac{\rho_v}{1-\rho_v} \lambda_d (1 - P_L)$, trong đó $\frac{\rho_v}{1-\rho_v} (1 - P_L)$ là số phương tiện liên kết với một RSU; λ_d là tham số tỉ lệ sinh gói tin ở mỗi xe. Tương tự, $\lambda_{V2N} = \frac{\rho_v}{1-\rho_v} \lambda_d N P_L$, $\lambda_{I2N} = \frac{\rho_v}{1-\rho_v} \lambda_d N (1 - P_L) P_R$. Vì vậy, số gói tin ở gNB có thể coi là tuân theo phân phối Poisson với tham số tỉ lệ là $\lambda_g = \frac{\rho_v}{1-\rho_v} \lambda_d N P_L + \frac{\rho_v}{1-\rho_v} \lambda_d N (1 - P_L) P_R$, trong đó $\frac{\rho_v}{1-\rho_v} \lambda_d N P_L$ là số gói tin được gửi bởi tất cả các xe, và $\frac{\rho_v}{1-\rho_v} \lambda_d N (1 - P_L) P_R$ là số gói tin được gửi bởi tất cả RSU.

Gọi λ_{I2V} là tỉ lệ gửi tin từ RSU về các xe, ta có $\lambda_{I2V} = \frac{\rho_v}{1-\rho_v} \lambda_d (1 - P_L) (1 - P_R)$. Giả sử rằng các gói tin là đồng nhất, ta coi p, q lần lượt là kích thước gói tin gửi và gói tin trả lời.

Loại 1: Các gói tin từ xe gửi trực tiếp lên gNB

Thời gian gửi một tác vụ từ một xe lên gNB được tính bởi công thức:

$$t_{V2N} = \frac{1}{\mu_{V2N} - \lambda_{V2N}} = \frac{1}{\mu_{V2N} - \frac{\rho_v}{1-\rho_v} \lambda_d N P_L} \quad (1)$$

Thời gian để giải quyết một tác vụ ở gNB là $t_g = \frac{1}{\mu_g - \lambda_g}$.

Thay λ_g vào công thức ta có:

$$t_g = \frac{1}{\mu_g - \frac{\rho_v}{1-\rho_v} \lambda_d N P_L - \frac{\rho_v}{1-\rho_v} \lambda_d N (1 - P_L) P_R} \quad (2)$$

Thời gian gửi gói tin trả lời từ gNB đến phương tiện được tính bởi $t_{N2V} = \frac{1}{\mu_{N2V} - \lambda_{N2V}}$, trong đó λ_{N2V} là tỉ lệ gói tin gửi từ gNB về các phương tiện trên đường. Tất cả các tác vụ được gửi từ RSU và các xe lên gNB sau đó sẽ được gửi lại về các xe, nên λ_{N2V} tương đương với λ_g . Cuối cùng ta có:

$$t_{N2V} = \frac{1}{\mu_{N2V} - \frac{\rho_v}{1-\rho_v} \lambda_d N P_L - \frac{\rho_v}{1-\rho_v} \lambda_d N (1 - P_L) P_R} \quad (3)$$

Từ (1), (2) và (3), tổng thời gian trễ của việc gửi, xử lý và nhận các gói tin thuộc loại 1 được biểu diễn bởi công thức:

$$t_1 = \frac{1}{\mu_{V2N} - \frac{\rho_v}{1-\rho_v} \lambda_d N P_L} + \frac{1}{\mu_g - \frac{\rho_v}{1-\rho_v} \lambda_d N P_L - \frac{\rho_v}{1-\rho_v} \lambda_d N (1 - P_L) P_R} + \frac{1}{\mu_{N2V} - \frac{\rho_v}{1-\rho_v} \lambda_d N P_L - \frac{\rho_v}{1-\rho_v} \lambda_d N (1 - P_L) P_R} \quad (4)$$

Loại 2: Các gói tin được gửi tới RSU và được xử lý trực tiếp tại RSU

Thời gian gửi tác vụ từ xe tới RSU là $t_{V2I} = \frac{1}{\mu_{V2I} - \lambda_{V2I}} = \frac{1}{\mu_{V2I} - \frac{\rho_v}{1-\rho_v} \lambda_d (1 - P_L)}$. Ta đã có μ_R đặc trưng cho khả năng phục

vụ của RSU, λ_R là tỉ lệ gói tin được gửi đến và xử lý tại RSU. Thời gian xử lý một tác vụ ở RSU được xác định như sau: $t_R = \frac{1}{\mu_R - \lambda_R}$. Do tỉ lệ gói tin được gửi tới RSU là

$\frac{\rho_v}{1-\rho_v} (1 - P_L) \lambda_d$ và xác suất RSU gửi tác vụ đó tới gNB là P_R , nên $\lambda_R = \frac{\rho_v}{1-\rho_v} \lambda_d (1 - P_L) (1 - P_R)$. Thời gian truyền

gói tin từ RSU đến các xe được tính bởi: $t_{I2V} = \frac{1}{\mu_{I2V} - \lambda_{I2V}} = \frac{1}{\mu_{I2V} - \frac{\rho_v}{1-\rho_v} \lambda_d (1 - P_L) (1 - P_R)}$. Vậy thời gian trễ trung

bình của việc xử lý và gửi nhận các tác vụ loại 2 được tính bởi:

$$t_2 = \frac{1}{\mu_{V2I} - \frac{\rho_v}{1-\rho_v}\lambda_d(1-P_L)} + \frac{1}{\mu_R - \frac{\rho_v}{1-\rho_v}\lambda_d(1-P_L)(1-P_R)} + \frac{1}{\mu_{I2V} - \frac{\rho_v}{1-\rho_v}\lambda_d(1-P_L)(1-P_R)} \quad (5)$$

Loại 3: Các tác vụ được gửi lên RSU rồi gửi lên gNB
Thời gian gửi từ RSU lên gNB là:

$$t_{I2N} = \frac{1}{\mu_{I2N} - \lambda_{I2N}} \quad (6)$$

Thay λ_{I2N} và λ_R vào (6), ta có:

$$t_{I2N} = \frac{1}{\mu_{I2N} - \frac{\rho_v}{1-\rho_v}\lambda_d N(1-P_L)P_R} \quad (7)$$

Tổng thời gian trễ của loại 3 được tính bằng: $t_3 = t_{V2I} + t_{I2N} + t_g + t_{N2V}$. Từ (7), (2), và (3), ta có:

$$t_3 = \frac{1}{\mu_{V2I} - \frac{\rho_v}{1-\rho_v}\lambda_d(1-P_L)} + \frac{1}{\mu_{I2N} - \frac{\rho_v}{1-\rho_v}\lambda_d N(1-P_L)P_R} + \frac{1}{\mu_g - \frac{\rho_v}{1-\rho_v}\lambda_d NP_L - \frac{\rho_v}{1-\rho_v}\lambda_d N(1-P_L)P_R} + \frac{1}{\mu_{N2V} - \frac{\rho_v}{1-\rho_v}\lambda_d NP_L - \frac{\rho_v}{1-\rho_v}\lambda_d N(1-P_L)P_R} \quad (8)$$

B. Phát biểu bài toán

Mục tiêu chính của bài toán là nhằm tối ưu tỉ lệ chuyển tác vụ từ các xe lên RSU và gNB và tối thiểu hóa thời gian trễ của cả mô hình. Xác suất của một gói tin tác vụ bất kỳ thuộc vào loại 1, loại 2, loại 3 tương ứng là $P_L, (1-P_L)(1-P_R)$ và $(1-P_L)P_R$. Do vậy, thời gian trễ trung bình của tất cả các tác vụ trong mô hình được xác định bởi công thức: $\bar{t} = P_L t_1 + (1-P_L)(1-P_R)t_2 + (1-P_L)P_R t_3$. Tổng hợp lại, bài toán tối ưu sẽ được phát biểu như sau:

Tối thiểu hóa

$$P_L t_1 + (1-P_L)(1-P_R)t_2 + (1-P_L)P_R t_3 \quad (9)$$

Sao cho thỏa mãn các ràng buộc dưới đây:

$$\frac{\rho_v}{1-\rho_v}(1-P_L)\lambda_d \leq \mu_{V2I} \quad (10)$$

$$\frac{\rho_v}{1-\rho_v}\lambda_d(1-P_L)(1-P_R) \leq \mu_{I2V} \quad (11)$$

$$\frac{\rho_v}{1-\rho_v}\lambda_d P_L N \leq \mu_{V2N} \quad (12)$$

$$\frac{\rho_v}{1-\rho_v}\lambda_d NP_L + \frac{\rho_v}{1-\rho_v}(1-P_L)\lambda_d NP_R \leq \mu_{N2V} \quad (13)$$

$$\frac{\rho_v}{1-\rho_v}\lambda_d(1-P_L)P_R N \leq \mu_{I2N} \quad (14)$$

$$\frac{\rho_v}{1-\rho_v}(1-P_L)(1-P_R)\lambda_d \leq \mu_R \quad (15)$$

$$\frac{\rho_v}{1-\rho_v}\lambda_d NP_L + \frac{\rho_v}{1-\rho_v}(1-P_L)\lambda_d NP_R \leq \mu_g \quad (16)$$

Ràng buộc (10), (11), (13) và (14) đảm bảo rằng tổng số tác vụ gửi không được vượt quá băng thông của mạng. Cụ thể (10) và (11) là ràng buộc băng thông giữa các xe và các RSU; (13) gắn với băng thông giữa xe và gNB; tương tự (14) đảm bảo băng thông giữa gNB và các xe đủ lớn. (15) và (16) là ràng buộc tỉ lệ gói tin tác vụ được xử lý bởi RSU và gNB không được vượt quá khả năng xử lý của chúng.

IV. TỐI ƯU HÓA XÁC SUẤT TRUYỀN TIN DỰA TRÊN GIẢI THUẬT DI TRUYỀN

Giải thuật di truyền là thuật toán tìm kiếm theo phương pháp heuristic dựa trên ý tưởng của chọn lọc tự nhiên và di truyền, được sử dụng phổ biến trong các bài toán tối ưu với hàm tối ưu phức tạp với kì vọng tạo ra lời giải có chất lượng cao. Giải thuật di truyền hiện nay đang được xem xét ứng dụng trong các bài toán tối ưu giảm tải trong mạng IOT [7] và mạng MEC [8], [9] vì tính linh hoạt, mặc dù thời gian hội tụ lâu nhưng có thể đáp ứng nhanh những yêu cầu với hàm tối ưu phức tạp cần đưa ra kết quả chấp nhận được trong thời gian tính toán hạn chế.

Trong chương này, chúng tôi đề xuất một giải thuật tối ưu dựa trên giải thuật di truyền (GA) để tìm lời giải tối ưu cho bài toán đã được đưa ra tại chương III.

A. Biểu diễn nhiễm sắc thể, hàm thích nghi, và khởi tạo quần thể

Một nhiễm sắc thể gồm hai gen biểu diễn hai giá trị P_L và P_R . Hàm thích nghi được tính bằng công thức (9). Mục tiêu của chúng tôi là xác định cá thể với độ thích nghi nhỏ nhất thỏa mãn tất cả các ràng buộc từ (10) đến (16). Chúng tôi khởi tạo quần thể một cách ngẫu nhiên. Để tăng độ đa dạng quần thể, chúng tôi tạo ra 70% cá thể thỏa mãn tất cả các ràng buộc từ (10) đến (16) và 30% còn lại vi phạm ít nhất một trong số các ràng buộc kể trên. Thêm vào đó, chúng tôi cũng sử dụng các giá trị biên (ví dụ: [0,0], [1,1], [0,1], [1,0]) để tăng thêm độ đa dạng của quần thể khởi tạo.

B. Lai ghép và đột biến

Chúng tôi đề xuất một giải thuật lai ghép trong đó sử dụng kết hợp 2 cách tiếp cận: *lai ghép trung bình* và *lai ghép điểm cắt*. Kí hiệu $I_1 = \{I_1.x, I_1.y\}$, $I_2 = \{I_2.x, I_2.y\}$ là hai cá thể bố mẹ, trong đó x và y biểu diễn giá trị của P_L và P_R theo thứ tự. Trong *lai ghép trung bình*, chúng tôi sinh ngẫu nhiên ra một số α có giá trị trong khoảng [0, 1] và sử dụng tham số α để làm trọng số kết hợp trung bình giữa 2 cá thể bố mẹ. Cụ thể, các gen của cá thể con $O = \{O.x, O.y\}$ được tính như sau: $O.x = \alpha I_1.x + (1-\alpha)I_2.x$ và $O.y = \alpha I_1.y + (1-\alpha)I_2.y$. Trong *lai ghép điểm cắt*, chúng tôi hoán đổi các gen giữa hai cá thể bố mẹ và chọn cá thể có độ thích nghi tốt hơn trong số hai cá thể con được sinh ra. Đầu tiên, chúng tôi tạo ra hai cá thể con $O_1 = \{I_1.x, I_2.y\}$ và $O_2 = \{I_2.x, I_1.y\}$, sau đó cá thể có giá trị hàm thích nghi nhỏ hơn sẽ được chọn cho thế hệ tiếp theo. Bằng việc sử dụng kết hợp cả hai phương pháp lai ghép trên, với một cặp bố mẹ, hai cá thể con sẽ được tạo ra sau phép lai ghép.

Với phép đột biến, chúng tôi đột biến mỗi cá thể bằng cách lấy trung bình cộng với các số được sinh ngẫu nhiên, có giá trị trong khoảng [0,1]. Cụ thể, coi $I = \{I.x, I.y\}$ là một cá thể cần đột biến, khi đó cá thể sau đột biến $O =$

$\{O.x, O.y\}$ được tạo ra như sau: $O.x = (x_0 + I.x)/2$ và $O.y = (y_0 + I.y)/2$. Trong đó x_0, y_0 là hai số ngẫu nhiên có giá trị nằm trong khoảng $[0,1]$.

C. Phương pháp chọn lọc và điều kiện dừng

Lưu ý rằng các cá thể con thu được sau khi thực hiện các phép lai ghép và đột biến có thể không thỏa mãn các ràng buộc (từ (10) đến (16)). Chúng tôi không loại bỏ những cá thể này khỏi quần thể, tuy nhiên gán cho chúng một độ thích nghi lớn đáng kể. Lý do là để duy trì độ đa dạng cao của quần thể. Sau khi thực hiện các phép lai ghép và đột biến, chúng tôi chọn ra M cá thể có giá trị hàm thích nghi thấp nhất để xây dựng quần thể thế hệ tiếp theo. Giải thuật dừng lại khi số lượng thế hệ đạt đến một giá trị ngưỡng đã được xác định.

D. Hạn chế của hướng tiếp cận GA

Phương pháp sử dụng giải thuật di truyền để xác định trước xác suất chuyển tác vụ P_R và P_L yêu cầu phải biết trước các thông số của mạng như: $\rho_v, \lambda_d, \mu_{V2I}, \mu_{V2N}, \mu_{I2N}, \dots$. Vì vậy nhược điểm của phương pháp tối ưu theo GA chúng tôi vừa trình bày có nhược điểm sau: (1) Trong thực tế, các phương tiện không tự nhiên biết trước các thông số cấu thành của mạng; (2) Vì các xác suất chuyển tác vụ P_R và P_L tính trước và cố định xuyên suốt quá trình hoạt động của phương tiện, nên GA không có khả năng tự điều chỉnh với các vấn đề phát sinh: gói tin bị hủy, một đường truyền quá tải,.... Với hai nhận xét trên, trong chương tiếp theo V, chúng tôi đề xuất giải pháp theo hướng tiếp cận học tăng cường. Với điểm mạnh là tính thích nghi cao, học tăng cường sẽ khắc phục được các nhược điểm của GA. Kết quả thí nghiệm ở phần VI chứng minh tính vượt trội hơn của phương pháp học tăng cường so với GA.

V. CƠ CHẾ TRUYỀN TIN ĐỘNG ỨNG DỤNG HỌC TĂNG CƯỜNG MAB

A. Multi-armed Bandits

Multi-Armed Bandits (MAB) là một kỹ thuật học tăng cường mà đã có nhiều ứng dụng thực tiễn với độ hiệu quả cao [10], [11], [12]. Trong một bài toán chuẩn tắc MAB, mỗi tác nhân (Agent) có K cánh tay (arms) hay được hiểu là K quyết định thực hiện các hành động khác nhau, ký hiệu $\mathcal{K} = 1, 2, \dots, K$. Một tác nhân sẽ thực hiện liên tiếp các hành động trong T vòng (Round), ký hiệu r_1, \dots, r_T . Một cánh tay $k \in \mathcal{K}$ thực hiện chỉ một hành động mà thu về cho tác nhân một giá trị số thực, tuân theo một phân phối \mathcal{P}_k với giá trị trung bình μ_k phản ánh độ tốt của hành động tương ứng tại vòng tương ứng mà hành động đó được thực hiện. Giá trị thực đó gọi là phần thưởng, ký hiệu R_t phần thưởng mà tác nhân thu về tại vòng thứ t . Mục tiêu của bài toán là chọn ra một hành động để thực hiện tại mỗi vòng $t \in \{1, 2, \dots, T\}$ để tối đa tổng cộng dồn phần thưởng: $\mathcal{X} = \sum_{t=1}^T R_t$. Một chuỗi hành động tối ưu như vậy là bất khả thi vào những thời gian đầu. Tuy nhiên, dựa vào tích lũy tri thức mà tác nhân có thể đưa ra những quyết định dựa vào tính toán xấp xỉ độ tốt của mỗi hành động và chọn được hành động có tiềm năng đem lại phần thưởng lớn nhất. Bài toán MAB miêu tả chính xác mâu thuẫn giữa hai hướng hành động đối lập tồn tại trong tất cả các bài toán ứng dụng phương pháp học tăng cường: khám phá và khai thác. Khai thác là quá trình mà tác nhân sẽ chọn hành động có phần thưởng khả kiến ở thời điểm hiện tại là lớn nhất. Trong khi

đó, khám phá là quá trình mà tác nhân sẽ chọn một hành động mà có thể không mang lại phần thưởng lớn nhất ở thời điểm hiện tại. Với hành động khám phá đó, tác nhân có thể nhận được tri thức mới, cải thiện đánh giá xấp xỉ các hành động hiện tại và tìm được các hành động mà cho phần thưởng cao hơn trong tương lai.

Trong bài nghiên cứu này, chúng tôi sử dụng MAB để giải quyết vấn đề độ trễ truyền tin với mạng ba tầng đa tác nhân, mô hình các RSU cũng như các phương tiện giao thông là các tác nhân của thuật toán. Chi tiết hơn, mỗi phương tiện giao thông có không gian hành động bao gồm: $RSU_offloading$ và $gNB_offloading$, tương ứng là gửi tác vụ lên RSU và gửi lên gNB. Đối với một RSU, không gian hành động bao gồm: $gNB_offloading$ và $local_processing$, lần lượt là chuyển tiếp tác vụ đến gNB và tự xử lý. Mục tiêu của bài toán là chọn được chuỗi các hành động sao cho độ trễ truyền tin trung bình của tất cả các tác vụ là nhỏ nhất. Hàm phần thưởng được định nghĩa trong mục V-C của bài, và trong mục V-D chúng tôi đề xuất một chiến lược khám phá mới và hiệu quả dựa trên hàm *Sigmoid*.

B. Tổng quan thuật toán

Trong chiến lược giảm tải của chúng tôi, mỗi phương tiện (hoặc RSU) duy trì một bảng hành động - độ tốt (viết tắt là *Bảng AV*). Bảng AV của mỗi phương tiện gồm 2 giá trị: RSU_value và gNB_value , mỗi giá trị tương ứng đại diện cho độ tốt của các hành động truyền tin đối với hành động chuyển gói tin lên RSU và gNB. Khi một phương tiện có tác vụ nào đó, nó sẽ sử dụng Bảng AV để quyết định sẽ gửi tác vụ đó cho RSU hay gNB. Tương tự, với mỗi RSU, Bảng AV bao gồm $local_value$ và gNB_value , trong đó $local_value$ thể hiện cho độ tốt của hành động xử lý bởi RSU và gNB_value cho hành động gửi lên gNB. Khi có một tác vụ gửi đến cho RSU, nó sẽ kiểm tra Bảng AV của mình và quyết định xem tác vụ đó sẽ được xử lý tại chỗ hay chuyển tiếp lên gNB. Khi một phương tiện (hay RSU) cần đưa ra quyết định truyền tin, một cơ chế ra quyết định (như mô tả ở phần V-D) sẽ được sử dụng, kết hợp với độ tốt của hành động được tính toán trong bảng AV để ra quyết định. Cụ thể, tác nhân hoặc hành động theo hướng khai thác hoặc theo hướng khám phá. Nếu sử dụng khai thác, tác nhân sẽ chọn hành động có độ tốt lớn nhất trong Bảng AV, ngược lại, nó sẽ chọn hành động ngẫu nhiên.

Độ tốt của hành động được xác định dựa trên độ trễ trung bình tích lũy của các tác vụ. Khi một phương tiện nhận được kết quả xử lý trả về từ RSU hoặc gNB, nó sẽ tính toán độ trễ và cập nhật độ tốt (giá trị trong Bảng AV) của hành động tương ứng. Các RSU không thể ước tính độ trễ này một cách trực tiếp vì các RSU không có thông tin về độ trễ của một tác vụ. Vì vậy, khi một phương tiện gửi một gói tin đến RSU, gói tin đó sẽ bao gồm thông tin về độ trễ của các gói tin trước đó. Chi tiết về Bảng AV sẽ được trình bày rõ hơn trong mục V-C.

C. Giảm tải với cơ chế thưởng âm

Thưởng âm (hay phạt, *negative rewarding* - NR) là một cơ chế khám phá của học tăng cường, được phát triển dựa trên cơ chế khởi tạo lạc quan (*Optimistic Initialization* - OI). Đối với cơ chế khám phá NR, thay vì khởi tạo giá trị xấp xỉ ban đầu của mỗi hành động là một số $L > \epsilon$ với $\epsilon = R_t, \forall t \in \{1, 2, 3, \dots, T\}$ và sử dụng giá trị phần thưởng trả về để cập nhật độ tốt của các hành động, NR khởi tạo giá trị

xấp xỉ ban đầu của mỗi hành động là 0 và tại mỗi vòng t ta sử dụng giá trị đối (âm) của phần thưởng để cập nhật độ tốt cho các hành động.

Chúng tôi xác định một hàm phần thưởng dựa trên độ trễ của mỗi loại tác vụ: loại 1, loại 2 và loại 3; lần lượt tương ứng với độ trễ trung bình của gói tin được truyền trực tiếp lên gNB, gói tin được RSU tiếp nhận và chuyển tiếp đến gNB và gói tin được RSU tiếp nhận sau đó xử lý. Khi phương tiện nhận được kết quả trả về của 1 tác vụ, nó sẽ tính toán độ trễ để hoàn thành tác vụ và sử dụng quy trình sau để xác định phần thưởng (theo cơ chế NR) và cập nhật Bảng AV:

- Với tác vụ loại 1, phương tiện sẽ lấy giá trị âm của độ trễ làm phần thưởng để cập nhật gNB_value cho hành động $gNB_offloading$ trong Bảng AV.
- Với tác vụ loại 2 hoặc loại 3, phương tiện lấy giá trị âm của độ trễ làm phần thưởng và cập nhật RSU_value tương ứng với hành động $RSU_offloading$.

Khi một phương tiện chuyển 1 tác vụ đến một RSU, gói tin đó sẽ mang theo thông tin trung bình độ trễ của các gói tin loại 2 và loại 3 tính kể từ lần gần nhất phương tiện đó truyền tin lên RSU. Ta kí hiệu \bar{t}_2, \bar{t}_3 , là độ trễ trung bình của các tác vụ loại 2 và loại 3 mà phương tiện gửi cho RSU. Khi nhận được thông tin này, RSU sẽ xác định phần thưởng cho các hành động tương ứng và cập nhật Bảng AV như sau:

- RSU sẽ lấy $-\bar{t}_2$ làm phần thưởng cho hành động $local_processing$ và cập nhật $local_value$ trong Bảng AV.
- RSU sẽ lấy $-\bar{t}_3$ làm phần thưởng cho hành động $gNB_offloading$ và cập nhật gNB_value trong Bảng AV.

Chúng tôi sử dụng công thức trung bình tích lũy để cập nhật Bảng AV như sau:

$$AV_t \leftarrow AV_t + \frac{R_t - AV_t}{N(t)} \quad (17)$$

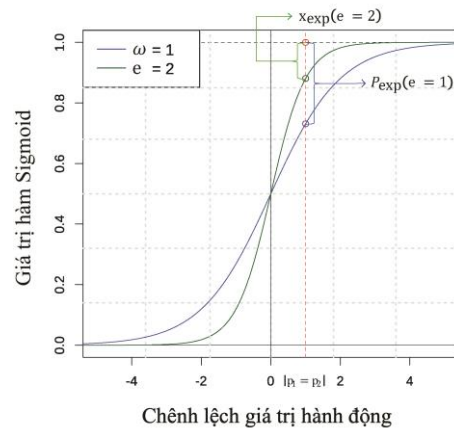
trong đó AV_t là giá trị của hành động, R_t là phần thưởng, $N(t)$ là số lần mà hành động đó đã được thực hiện. Ví dụ, giả sử một phương tiện đã chuyển n_1 tác vụ đến các RSU và n_2 tác vụ đến gNB. Khi phương tiện đó nhận lại một gói tin kết quả thuộc loại 1, phương tiện sẽ tính toán độ trễ Δ_t và cập nhật gNB_value bằng công thức (17):

$$gNB_value \leftarrow gNB_value + \frac{-\Delta_t - gNB_value}{n_2}$$

D. Cơ chế khám phá Sigmoid (SEM)

Trong phần này chúng tôi giới thiệu một cơ chế khám phá mới, điều chỉnh xác suất khám phá dựa trên kinh nghiệm. Cụ thể, ta sẽ giảm xác suất khám phá khi mà độ chênh lệch giá trị giữa hai hành động đủ lớn và tăng trong trường hợp ngược lại. Lí do ta làm vậy là khi độ chênh lệch giá trị giữa hai hành động đủ lớn, điều đó chứng tỏ tồn tại một hành động có giá trị lớn hơn và thực sự tốt hơn hành động còn lại, nên ta sẽ duy trì hành động khai thác và giảm xác suất khám phá. Ngược lại, khi chênh lệch này tương đối nhỏ, có nghĩa là 2 hành động không có sự khác biệt lớn về độ tốt, ta sẽ tăng xác

suất khám phá lên để tác nhân có thể tìm hiểu môi trường tốt hơn thay vì tập trung vào một hành động tốt nhất hiện tại. Chúng tôi sẽ trình bày chi tiết công thức tính xác suất khám phá cho mô hình V2X ở bên dưới.



Hình 2: Xác suất khám phá dựa trên hàm Sigmoid

Kí hiệu r_1 và r_2 là giá trị (hay độ tốt) của hai hành động của phương tiện (hay RSU). Khi đó, xác suất khám phá được tính dựa trên hàm sigmoid như sau:

$$P_{exp} = 1 - \frac{1}{1 + e^{-\omega|r_1 - r_2|}} \quad (18)$$

Chúng tôi ứng dụng hàm sigmoid bởi vì giá trị của đầu ra luôn nằm trong khoảng từ 0 đến 1, phù hợp để thể hiện giá trị của xác suất. Hơn nữa, giá trị của $|r_1 - r_2|$ càng lớn thì giá trị của P_{exp} càng nhỏ, phù hợp với lập luận đã nêu. Bên cạnh đó độ dốc của hàm có thể điều chỉnh bằng tham số ω . Hình: 2 miêu tả đồ thị của hàm tỉ lệ khám phá sigmoid khi ta biến thiên ω . Chúng ta có thể thấy, với cùng giá trị của $r_1 - r_2$, tăng giá trị ω sẽ làm giảm xác suất khám phá. Hơn nữa, độ dốc của hàm sigmoid cũng tăng lên trong trường hợp này. Giảm giá trị ω sẽ làm mượt hàm sigmoid hơn. Khi $\omega = 0$, xác suất khám phá sẽ luôn là hằng số và không phụ thuộc vào giá trị của $r_1 - r_2$. Ngược lại khi ω đủ lớn, P_{exp} sẽ tiệm cận bằng 0, tác nhân sẽ không thực hiện khám phá và trở về với cơ chế NR thông thường.

Bảng 1: Các thông số mô phỏng

Thông số	Giá trị
Xung nhịp CPU của gNB	256 Ghz
Xung nhịp CPU của RSU	64 Ghz
Xung nhịp CPU cần cho mỗi tác vụ	0.2 Ghz
Kích thước trung bình gói tin	500 kb
Băng thông truyền RSU và gNB	10 Gbps
Băng thông truyền phương tiện và RSU	1 Gbps
Băng thông truyền phương tiện và gNB	500 Mbps
Khoảng chờ xuất hiện phương tiện ($\frac{1}{\lambda_v}$)	5 ~ 9
Tần suất sinh tác vụ (λ_d)	70 ~ 100
Chiều dài đường	1500 m
Tốc độ của phương tiện	12 m/s

E. Độ phức tạp tính toán

Đối với hướng tiếp cận GA, chúng tôi xác định trước hai biến xác suất P_L và P_R , trong quá trình mô phỏng không bao gồm các tính toán nào khác. Chúng tôi khởi tạo quần thể số lượng n cá thể, tại mỗi chu kỳ, ta duyệt qua quần thể, mỗi cá thể gồm 2 gen với độ phức tạp tính toán là $O(n)$. Tương tự với phép đột biến và chọn lọc cũng có độ phức tạp $O(n)$. Tóm lại, phương pháp GA có độ phức tạp tính toán là $O(n + n + n) = O(n)$ đối với một chu kỳ lai ghép. Đối với hướng tiếp cận học tăng cường, ta thực hiện các cập nhật ở mỗi lần tác nhân hoàn tất một hành động. Cụ thể, trong bài, bảng AV của tác nhân gồm có hai hành động, mỗi lần cập nhật chỉ một hành động, do vậy tốn $O(1)$ vì giá trị các hành động được cập nhật theo công thức 17. Khi đưa ra quyết định, ta sử dụng cơ chế SEM giữa hai hành động này áp dụng công thức 18, độ phức tạp tính toán là $O(1)$.

VI. KẾT QUẢ

A. Cài đặt thí nghiệm

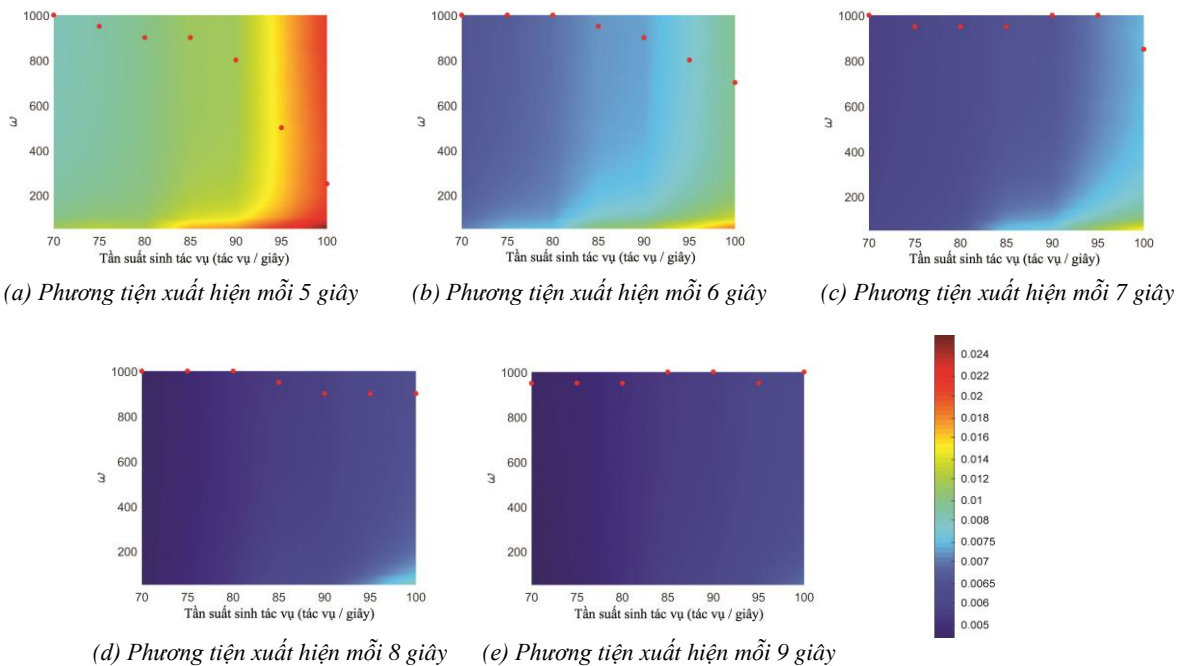
Trong chương này, đầu tiên, chúng tôi sẽ phân tích ảnh hưởng của ω đến cơ chế khám phá SEM, và so sánh SEM với cơ chế khám phá *thường âm* (còn gọi là cơ chế phạt,

B. Ảnh hưởng của ω và so sánh SEM với NR

Trong phần này, chúng tôi nghiên cứu sự ảnh hưởng của ω đến cơ chế SEM và so sánh SEM với NR.

Đầu tiên, chúng tôi thay đổi tần suất xuất hiện của các phương tiện và tần suất sinh gói tin của mỗi phương tiện, từ đó tìm hiểu mối quan hệ giữa giá trị tối ưu ω đối với sự thay đổi đó. Hình 3 gồm các biểu đồ nhiệt thể hiện độ trễ truyền tin trung bình khi ta thay đổi tần suất sinh gói tin và tần suất xuất hiện của các phương tiện. Ta quan sát được rằng, độ trễ truyền tin tăng lên cùng với sự tăng lên của tần suất xuất hiện phương tiện và tần suất sinh gói tin. Hơn nữa, giá trị tối ưu của ω cũng có xu hướng giảm khi ta tăng tần suất xuất hiện gói tin. Quan sát này được thể hiện rõ nhất tại Hình 3(a) và Hình 3(b). Bằng cách so sánh các biểu đồ nhiệt với nhau, ta cũng nhận ra rằng giá trị tối ưu của ω có xu hướng giảm khi tần suất xuất hiện của các phương tiện giao thông tăng lên. Lý do là vì sự tăng lên của tần suất xuất hiện các phương tiện giao thông, mật độ phương tiện trên đường tăng lên, số lượng gói tin yêu cầu xử lý lớn hơn, nên cần phải có cơ chế khám phá thường xuyên để cập nhật các thông tin của mạng nhanh hơn, để thay đổi hành động tối ưu phù hợp với tốc độ sinh gói tin.

Tiếp theo, chúng tôi đặt cố định tần suất xuất hiện của



Hình 3: Tác động của ω đến độ trễ trung bình (các chấm màu đỏ mô tả các giá trị tối ưu của ω)

negative rewarding - NR). Sau đó, chúng tôi so sánh SEM với GA trình bày tại chương IV. Trong trình giả lập thí nghiệm, các gói tin xuất hiện và xử lý theo phân phối Poisson. Ngoài ra, các thí nghiệm chạy với các gói tin được cài đặt để xuất hiện theo phân phối ngẫu nhiên. Các thông số cài đặt còn lại được cho trong bảng I. Hơn nữa, các RSU được phân bố đều theo bề ngang, dọc theo đường đi. Chúng tôi thí nghiệm với các kịch bản có số lượng RSU thay đổi từ 5 đến 9, đồng thời tần suất sinh gói tin theo phân phối thay đổi từ 70 gói tin/giây đến 100 gói tin/giây và tần suất phương tiện xuất hiện từ 5 giây/phương tiện (mật độ dày) đến 9 giây/phương tiện (mật độ thưa).

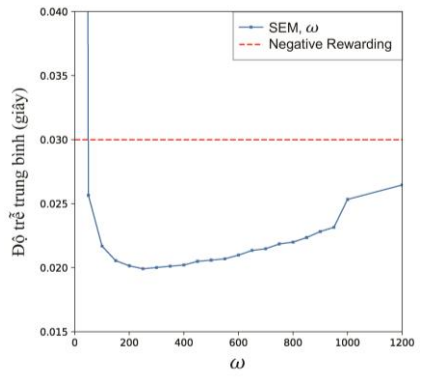
các phương tiện là 5 giây/phương tiện và cố định giá trị trung bình λ_d của phân phối sinh gói tin là 100 rồi thay đổi các giá trị của ω từ 1 đến 1200 và dựng đồ thị độ trễ trung bình và độ trễ trung bình tích lũy trong Hình 4. Trong Hình 4(a), trục x là trục giá trị của ω , trục y là trục độ trễ trung bình. Đồ thị cho thấy rằng khi ta tăng giá trị của ω từ 1 đến 250, độ trễ trung bình giảm dần. Bởi vì, giá trị nhỏ của ω sẽ làm tỉ lệ khám phá cao, nên khi ω được đặt ở giá trị tương đối nhỏ thì phương tiện truyền tin một cách gần như ngẫu nhiên, gây ra bởi tỉ lệ khám phá quá cao. Do đó, khả năng cao là các quyết định được đưa ra không dựa trên đánh giá khách quan của độ trễ mà do các gói tin trong

mạng. Tăng giá trị ω sẽ giúp cho tác nhân khai thác được hành động có phần thưởng lớn và vì vậy, giảm được độ trễ của các gói tin. Tuy nhiên, Hình 4(a), cũng chỉ ra rằng, từ sau giá trị $\omega = 250$, độ trễ trung bình tăng dần. Điều này là do tỉ lệ khám phá thấp dẫn đến sự thiếu hụt tri thức cần thiết để tìm ra hành động có độ tốt cao hơn, nên tác nhân tốn nhiều thời gian để tìm được hành động phù hợp, dẫn đến độ trễ trung bình cao hơn. Độ trễ trung bình tích lũy được biểu diễn bằng đồ thị ở Hình 4(b), Trong đồ thị này, trục x là trục của thời gian trong khi trục y là trục của độ trễ trung bình tích lũy của các gói tin. Đồ thị này thể hiện tính ổn định của các lựa chọn cho giá trị của ω , các lựa chọn ω nhỏ hơn có độ ổn định thấp hơn bởi tỉ lệ khám phá cao hơn.

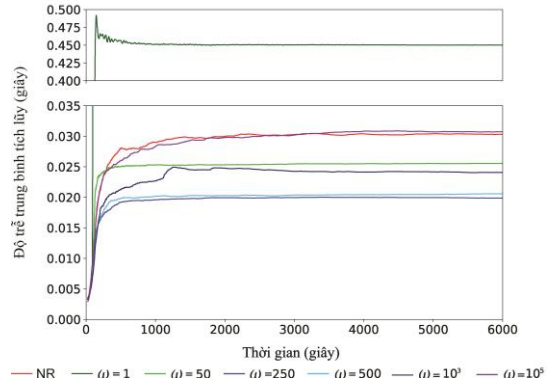
Bảng II so sánh sự khác biệt giữa việc ra quyết định theo cơ chế SEM và cơ chế NR. Ta xem xét cụ thể một phương tiện \mathcal{V} đang di chuyển và một RSU \mathcal{R} , giả sử $\Delta_{\mathcal{V}}$ là chênh lệch của độ tốt hiện tại giữa hai hành động của \mathcal{V} , cụ thể

$\Delta_{\mathcal{V}} = gNB_value - RSU_value$; $\Delta_{\mathcal{R}}$ là chênh lệch của độ tốt hiện tại giữa hai hành động của \mathcal{R} , cụ thể, $\Delta_{\mathcal{R}} = gNB_value - local_value$. Trong NR, các tác nhân luôn lựa chọn hành động đem lại "ít điểm phạt" hơn, vì vậy, \mathcal{V} và \mathcal{R} luôn chọn truyền tin tới gNB trong trường hợp $\Delta_{\mathcal{V}}$ và $\Delta_{\mathcal{R}}$ nhận giá trị dương. Trái lại, cơ chế của SEM lại linh động và linh động có định hướng. Ví dụ, nếu một phương tiện \mathcal{V} sử dụng cơ chế SEM, tỉ lệ chuyển gói tin lên gNB phụ thuộc không chỉ vào dấu của $\Delta_{\mathcal{V}}$ mà còn phụ thuộc vào giá trị của nó. Cụ thể, giả sử $\Delta_{\mathcal{V}} = 0.00302$, thì tỉ lệ gói tin được chuyển lên gNB là 95.25%, trong khi với giá trị $\Delta_{\mathcal{V}} = 0.0011$ (nhỏ hơn khoảng 3 lần) thì tỉ lệ này là 75.03%.

Hình 4(a) so sánh độ trễ trung bình của các gói tin theo hai cơ chế khám phá NR và SEM. Như biểu đồ cho thấy, với giá trị phù hợp của ω (ví dụ $\omega = 200; 250$), cơ chế SEM giảm độ trễ trung bình một lượng lớn so với NR. Cụ thể, với $\omega = 250$, độ trễ trung bình theo cơ chế SEM chỉ



(a) Độ trễ trung bình



(b) Độ trễ tích lũy

Hình 4: Độ trễ của tác vụ liên quan đến các cài đặt của sơ đồ SEM và Thưởng âm (NR)

Bảng II: So sánh truyền tin giữa NR và SEM ($\omega = 1000$)

$\Delta_{\mathcal{V}}$	$\Delta_{\mathcal{R}}$	Thưởng âm (NR)	SEM
0.00302	0.0012	100% loại (1)	95.25% loại (1), 1.1% loại (2), 3.65% loại (3)
0.0011	-0.026	100% loại (1)	75.03% loại (1), 24.97% loại (2), 1.28% loại (3)
-0.0125	0.0025	100% loại (3)	0.0004% loại (1), 7.5857% loại (2), 92.4138% loại (3)
-0.0005	-0.0005	100% loại (2)	37.7541% loại (1), 38.7455% loại (2), 23.5004% loại (3)

Bảng III: Cấu hình xác suất truyền tin

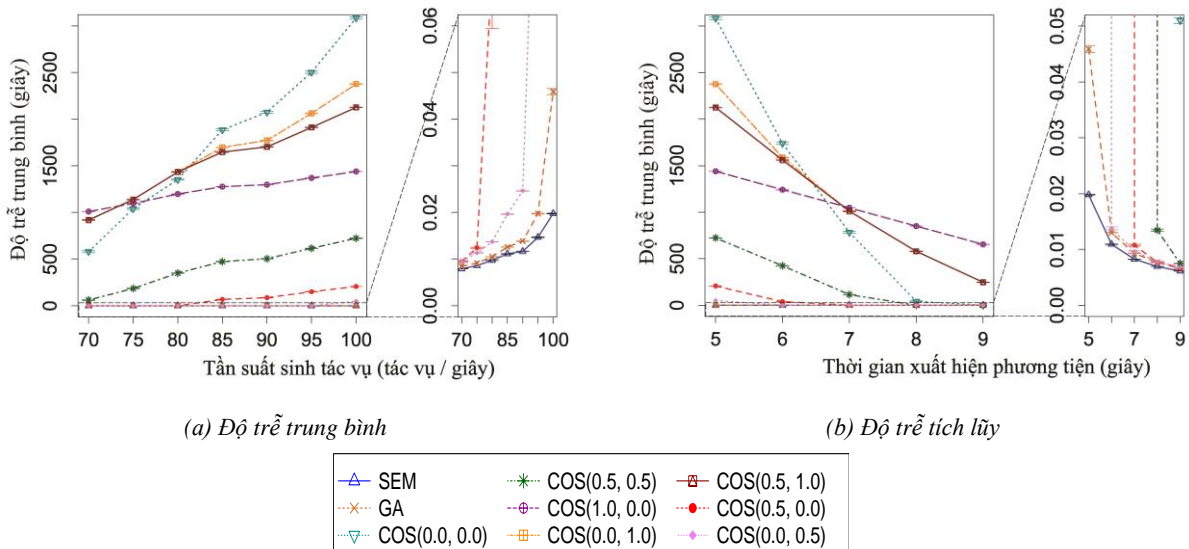
P_L	P_R	Mô hình	Ý nghĩa
0	0	2 tầng	Tất cả các tác vụ được xử lý tại RSU.
1	0	2 tầng	Tất cả các tác vụ được xử lý tại gNB.
0	0.5	3 tầng	Phương tiện truyền tất cả các tác vụ lên RSU. RSU truyền 50% tác vụ lên gNB.
0	1	3 tầng	Phương tiện truyền tất cả các tác vụ lên RSU. RSU truyền tất cả các tác vụ lên gNB.
0.5	0	3 tầng	Phương tiện truyền 50% tác vụ lên gNB, 50% tác vụ lên RSU; RSU tự xử lý tất cả các tác vụ.
0.5	0.5	3 tầng	Phương tiện truyền 50% tác vụ lên gNB, 50% tác vụ lên RSU; RSU truyền 50% tác vụ lên gNB.
0.5	1	3 tầng	Phương tiện truyền 50% tác vụ lên gNB, 50% tác vụ lên RSU; RSU truyền tất cả tác vụ lên gNB.

bằng 65% độ trễ theo NR. Với giá trị rất lớn của ω , tức tỉ lệ khám phá theo SEM tiệm cận về 0, thì SEM có kết quả tương đương với NR (Hình 4(b)). Không chỉ giảm được độ trễ trung bình, SEM còn có thời gian hội tụ nhanh hơn nhiều so với NR, cụ thể với $\omega = 250$: SEM hội tụ và ổn định từ thời điểm 2000s giây sau khi bắt đầu giả lập, trong khi NR cần hơn 5000s.

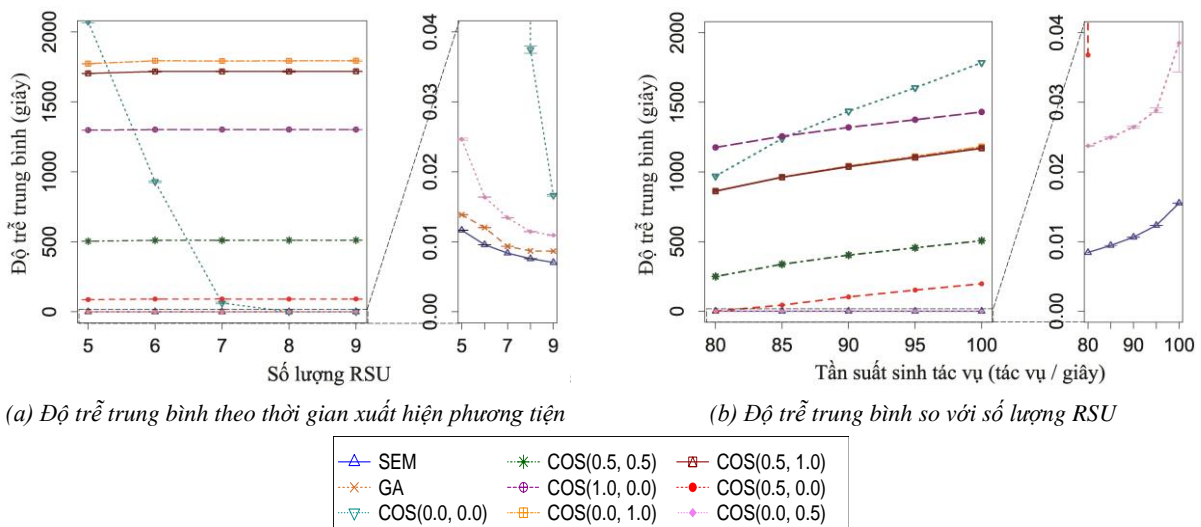
C. SEM so sánh với GA, COS

Trong phần này, chúng tôi so sánh kết quả của cơ chế SEM với phương pháp GA và phương pháp sử dụng cố định tỉ lệ truyền tin (viết tắt là COS). Các cài đặt của tỉ lệ truyền tin P_R và P_L của COS được trình bày trong Bảng III. Chúng tôi thực hiện các thí nghiệm để tìm hiểu kết quả của cả ba hướng tiếp cận bằng cách biến thiên ba yếu tố chủ đạo: tần suất sinh gói tin của mỗi phương tiện, tần suất phương tiện xuất hiện trên đường và số lượng đơn vị điện toán biên (RSU) trên quãng đường đó. Các thông số đo lường chúng tôi dùng để đánh giá kết quả bao gồm độ trễ trung bình của tất cả các gói tin và số lượng gói tin bị hủy.

1) Average latency: Trong thí nghiệm đầu tiên, chúng tôi đặt 5 RSU trên đường, cố định tần suất xuất hiện của các phương tiện là 5 gói/phương tiện, và chạy giả lập với số lượng gói tin trung bình sinh ra tại mỗi phương tiện giao thông thay đổi từ 70 đến 100. Các gói tin được sinh ra theo hai phân phối: phân phối Poisson và phân phối ngẫu nhiên. Kết quả của thí nghiệm lần lượt được biểu diễn ở Hình: 5(a) và 5(b) Trong bảng ký hiệu, $COS(a, b)$ được định nghĩa là sử dụng cơ chế COS với $P_L = a$ và $P_R = b$. Và vì GA chỉ ứng dụng cho phân phối Poisson biết trước trung bình nên kết quả của GA không xuất hiện ở Hình: 5(b). Ta có thể quan sát được rằng, dù với cơ chế truyền tin nào thì xu hướng chung là độ trễ trung bình tăng lên khi tần suất sinh gói tin của các phương tiện giao thông tăng lên. Hiện tượng này là hiển nhiên bởi tần suất sinh gói tin lớn đặt khối lượng xử lý nặng hơn lên các đường truyền và lên các đơn vị xử lý, kết quả là độ trễ trung bình của các gói tin tăng lên. Trong tất cả các trường hợp, SEM có độ trễ trung bình là nhỏ nhất và nhỏ hơn rất nhiều so với hai hướng tiếp cận còn lại. COS cho kết quả tồi nhất đặc biệt trong môi trường mạng khắc nghiệt (tần suất sinh gói tin cao, tần suất xuất hiện phương tiện cao, số lượng RSU thấp). GA có độ



Hình 5: So sánh độ trễ trung bình theo các tỉ lệ sinh ra tác vụ khác nhau



Hình 6: Ảnh hưởng của tần suất xuất hiện xe và số lượng RSU

Thí nghiệm	Tham số	SEM	GA	COS						
				(0, 0)	(0.5, 0.5)	(1, 0)	(0, 1)	(0.5, 1)	(0.5, 0)	(0, 0.5)
# 1	Tần suất sinh gói tin									
	70	0	1.9×10^{-7}	97.44	0.087	0	0	0	0.177	0.177
	75	0	3.2×10^{-6}	98.43	0.095	0	0	0	0.189	0.189
	80	0	3.6×10^{-6}	98.93	0.101	0	0	0	0.205	0.205
	85	2.8×10^{-7}	5.2×10^{-6}	99.16	0.1	0	0	0	0.202	0.196
	90	4.5×10^{-7}	2.7×10^{-5}	99.22	0.08	0	0	0	0.163	0.156
	95	6.2×10^{-6}	1.2×10^{-4}	99.4	0.09	0	0	0	0.194	0.181
100	2×10^{-5}	7.8×10^{-4}	99.48	0.082	0	0	0	0.169	0.153	
# 2	Khoảng đợi phương tiện mới									
	6	7.7×10^{-7}	$1.9.10^{-4}$	99.33	0.94	0	0	0	1.879	1.877
	7	0	10^{-6}	87.18	1.031	0	0	0	2.064	2.067
	8	0	10^{-6}	61.63	0.935	0	0	0	1.867	1.867
	9	0	0	22.6	1.031	0	0	0	2.062	2.063
# 3	Số lượng RSU									
	6	0	0	99.15	1.065	0	0	0	2.131	2.137
	7	0	0	63.49	0.7823	0	0	0	1.566	1.567
	8	0	0	18.16	0.554	0	0	0	1.116	1.115
	9	0	0	16.63	0.42	0	0	0	0.841	0.842

Bảng IV: So sánh tỉ lệ mất gói tin

trễ trung bình nhỏ hơn đáng kể so với COS. Tuy nhiên, vì phụ thuộc hoàn toàn vào tỉ lệ truyền tin đã tính trước, GA thất bại trong việc xử lý các tình huống phát sinh như gói tin bị hủy hoặc một đường truyền bị nghẽn. Ta có thể thấy rằng kết quả của GA và SEM tách biệt rõ ràng hơn khi môi trường mạng khắc nghiệt hơn. Hơn nữa, độ dốc của đồ thị độ trễ trung bình của SEM cũng nhỏ hơn rất nhiều so với các phương pháp còn lại. SEM giảm độ trễ truyền tin trung bình 18.5% , trong trường hợp tốt nhất là 56.9% giầy so với GA. So với cài đặt tốt nhất của COS, cụ thể là $P_L = 0, P_R = 0.5$ thì kết quả của SEM thấp hơn từ 16.2% tới 99.9% trong trường hợp gói tin sinh theo phân phối Poisson và từ 57.3% tới 99.9% trong trường hợp phân phối ngẫu nhiên.

Trong thí nghiệm thứ hai, chúng tôi cố định tần suất sinh gói tin ở mức 100 gói tin/giây, sử dụng 5 RSU, và sau đó biến thiên tần suất xuất hiện của phương tiện giao thông từ 5 giây/phương tiện lên thành 9 giây/phương tiện. Hình 6(a) thể hiện độ trễ trung bình của SEM, GA và các cài đặt của COS. Ta nhận thấy rằng SEM có kết quả tốt hơn cả hai hướng tiếp cận còn lại trong tất cả các thí nghiệm. Hơn nữa, kết quả của SEM tốt hơn một cách vượt trội trong các thí nghiệm với lượng phương tiện dày đặc. Với tần suất xuất hiện của phương tiện lớn hơn 5 giây/phương tiện, thì độ trễ trung bình của các gói tin ở mức khoảng 90% độ trễ của GA. Con số này giảm xuống còn 56.9% khi tần suất xuất hiện của phương tiện cao: 5 giây/phương tiện. Đối với cài đặt tốt nhất của COS, SEM giảm độ trễ trung bình ít nhất là 10.2%. Đặc biệt trong trường hợp mật độ phương tiện lớn, SEM giảm độ trễ trung bình đến 99,9%. Những kết quả này khẳng định một lần nữa sự vượt trội của cơ chế SEM đối với môi trường mạng khắc nghiệt.

Trong thí nghiệm thứ ba, chúng tôi tìm hiểu ảnh hưởng của số lượng RSU đối với độ trễ trung bình trong các trường hợp khác nhau của tần suất sinh gói tin và tần suất xuất hiện phương tiện trong mạng. Kết quả được biểu diễn tại Hình 6(b), trong đó, COS(0,0) bị ảnh hưởng nhiều nhất vì COS(0,0) gửi toàn bộ gói tin của phương tiện đến các RSU và xử lý tại RSU. Do đó, tăng số lượng RSU sẽ giảm được độ trễ của cả đường truyền và xử lý. Độ trễ trung bình cũng giảm khi tăng số lượng RSU đối với SEM, GA và COS(0,0.5). So với GA, SEM giảm được độ trễ trung bình là 15.6%, trong trường hợp tốt nhất là 20.3%. So với COS,

kết quả của SEM tốt hơn so với cài đặt tốt nhất của COS giao động trong khoảng 33.8% đến 52.7%.

2) *Tỉ lệ gói tin bị hủy*: Bảng IV cho thấy tỉ lệ gói tin bị hủy của các cơ chế truyền tin. Một gói tin bị hủy khi một phương tiện di chuyển ra khỏi phạm vi phủ sóng của RSU đang tương tác mà chưa nhận lại gói tin phản hồi từ RSU đó. Chúng ta có thể thấy rằng nếu sử dụng cơ chế COS với cài đặt hoặc $P_R = 1$ hoặc $P_L = 1$ đều có tỉ lệ hủy gói tin bằng 0. Bởi các cài đặt này luôn đảm bảo gói tin được xử lý tại gNB, và vì vậy, luôn được gửi lại cho phương tiện giao thông ban đầu, tuy nhiên, như đã đề cập trong phần trước, các cài đặt này chịu một lượng trễ truyền tin vô cùng lớn, kết quả là độ trễ trung bình của các gói tin cao hơn rất nhiều. Ta có thể thấy rằng, SEM là cơ chế cho kết quả gói tin bị hủy ít thứ hai (sau ba cài đặt của COS). Với tần suất sinh gói tin không lớn hơn 80 gói tin/giây và tần suất xuất hiện phương tiện không cao hơn hơn 7 giây/phương tiện, số lượng RSU trên đường không bé hơn 6 thì SEM có tỉ lệ gói tin bị hủy là 0%. Trong trường hợp mạng chịu tải nặng (tần suất sinh gói tin cao, tần suất phương tiện xuất hiện cao) thì SEM vẫn có kết quả vượt trội hơn so với GA. Cụ thể, ngoài các trường hợp đạt tỉ lệ truyền tin 100% thì SEM có tỉ lệ hủy gói tin thấp hơn GA khoảng 10 đến 100 lần.

VII. HƯỚNG NGHIÊN CỨU TIẾP THEO

Bài báo này nghiên cứu vấn đề tối thiểu hóa độ trễ truyền tin trong mạng ba tầng V2X. Chúng tôi đề xuất một chiến lược truyền tin mới, ứng dụng kỹ thuật học tăng cường multi-agents multi-armed bandits (MAB) để đưa ra quyết định thời gian thực, thích nghi với môi trường mạng động và thiếu ổn định. Chiến lược chúng tôi đề xuất được tích hợp với cơ chế khám phá dựa trên hàm Sigmoid, cho kết quả tốt hơn đáng kể với cơ chế khám phá thông thường. Hơn nữa, kết quả thí nghiệm đã chứng minh tính hiệu quả trong việc giảm thiểu độ trễ trung bình của mỗi tác vụ trung bình 18.5%, trong trường hợp tốt nhất là 56.9% so với hướng tiếp cận sử dụng giải thuật di truyền (GA).

Trong tương lai, chúng tôi mở rộng giải thuật MAB vào mô hình mạng ba tầng hỗ trợ các đường truyền I2I (giữa các RSU) và V2V (giữa các phương tiện).

LỜI CẢM ƠN

Nghiên cứu này được tài trợ bởi Quỹ Phát triển khoa học và công nghệ Quốc gia (NAFOSTED) trong đề tài mã số 102.01-2019.302.

Một phần nghiên cứu này đã được trình bày trong công bố [13]: Phi Le Nguyen, Ren-Hung Hwang, Phạm Minh Khiêm, Kien Nguyen, Ying-Dar Lin, “Modeling and Minimizing Latency in Three-tier V2X Networks”, The 2020 IEEE Global Communications Conference (IEEE GLOBECOM), 8–10 December 2020, Taipei, Taiwan.

TÀI LIỆU THAM KHẢO

- [1] L. Feng and W. Li and Y. Lin and L. Zhu and S. Guo and Z. Zhen, “Joint Computation Offloading and URLLC Resource Allocation for Collaborative MEC Assisted Cellular-V2X Networks,” trong *IEEE Access*, 2020.
- [2] G. Wang và F. Xu, “Regional Intelligent Resource Allocation in Mobile Edge Computing Based Vehicular Network,” trong *IEEE Access*, 2020.
- [3] X. and Cui, T. and Cao, C. and Chen, Q. and Kwak, K.S. Fan, “Minimum-Cost Offloading for Collaborative Task Execution of MEC-Assisted Platooning},,” *Sensors*, tập 19, 2019.
- [4] H. Wang and X. Li and H. Ji and H. Zhang, “Federated Offloading Scheme to Minimize Latency in MEC-Enabled Vehicular Networks,” trong *Proc. IEEE GLOBECOM Workshops*, 2018.
- [5] Cui, T. and Hu, Y. and Shen, B. and Chen, Q., “Task Offloading Based on Lyapunov Optimization for MEC-Assisted Vehicular Platooning Networks,” *Sensors*, tập 19, 2019.
- [6] J. Zhao and Q. Li and Y. Gong and K. Zhang, “Computation Offloading and Resource Allocation For Cloud Assisted Mobile Edge Computing in Vehicular Networks,” *IEEE Trans. Veh. Technol.*, tập 68, số 8, pp. 7944-7956, 2019.
- [7] Hussain, Azham and Manikanthan, Sv and T., Padmapriya and Nagalingam, Mahendran, “Genetic algorithm based adaptive offloading for improving IoT device communication efficiency,” *Wireless Networks*, 2020.
- [8] Liao, Zhuofan and Peng, Jingsheng and Xiong, Bing and Huang, Jiawei, “Adaptive offloading in mobile-edge computing for ultra-dense cellular networks based on genetic algorithm,” *Journal of Cloud Computing*, tập 10, 2021.
- [9] Li, Zhi and Zhu, Qi, “Genetic Algorithm-Based Optimization of Offloading and Resource Allocation in Mobile-Edge Computing,” *Information*, tập 11, số 2, 2020.
- [10] S. J. Darak and M. K. Hanawal, “Multi-Player Multi-Armed Bandits for Stable Allocation in Heterogeneous Ad-Hoc Networks,” *IEEE J. Sel. Areas Commun.*, tập 37, số 10, pp. 2350-2363, 2019.
- [11] A. Verma and M. K. Hanawal, “Stochastic Network Utility Maximization with Unknown Utilities: Multi-Armed Bandits Approach,” *Proc. IEEE INFOCOM*, pp. 189-198, 2020.
- [12] Moerchen, Fabian and Ernst, Patrick and Zappella, Giovanni, “Personalizing Natural Language Understanding Using Multi-Armed Bandits and Implicit Feedback,” trong *ACM CIKM 2020*, 2020, p. 2661-2668.
- [13] P. L. Nguyen and R. -H. Hwang and P. M. Khiem and K. Nguyen and Y. -D. Lin, “Modeling and Minimizing Latency in Three-tier V2X Networks,” trong *Proc. GLOBECOM 2020*, 2020.

REINFORCEMENT LEARNING-BASED OFFLOADING ALGORITHM FOR DELAY MINIMIZATION IN 3-TIER VEHICULAR NETWORKS

Abstract—Exploiting Mobile Cloud Computing (MCC) and Mobile Edge Computing (MEC) to offload computational tasks to servers is a promising approach that enables delay-sensitive applications in vehicular networks. Most of the existing work focuses on using only either MCC or MEC. This research studies a combination of the two technologies and proposes a three-tier V2X network, where vehicles can offload computational tasks using MEC and MCC. Specifically, we investigate the optimal offloading probabilities over three communication paths: V2I, V2N, and I2N. To this end, we first propose a mathematical model formulating the average delay and a theoretical formulation determining the exact optimal solution. We then present a genetic algorithm-based algorithm to determine the sub-optimal solution. Finally, we apply reinforcement learning techniques that can adapt to the network dynamics in making the offloading decision. The experimental results show that the GA-based algorithm can shorten the latency up to 99.9% compared with the existing two-tier models. Furthermore, using reinforcement learning can reduce latency by at least 18.5% compared to the GA-based algorithm.

Keywords—Reinforcement Learning, MEC, V2X.



Nguyễn Năng Hưng hiện tại là sinh viên năm thứ ba tại trường Công nghệ thông tin và Truyền thông, Đại học Bách Khoa Hà Nội. Đồng thời anh cũng đang là trợ lý nghiên cứu tại phòng thí nghiệm Intelligent Communication Networks, trung tâm BK.AI. Nghiên cứu của anh liên quan tới tối ưu, học tăng cường và IoT.



IoT.

Nguyễn Trung Thành hiện tại là sinh viên năm thứ tư tại trường Công nghệ thông tin và Truyền thông, Đại học Bách Khoa Hà Nội. Đồng thời anh cũng đang là trợ lý nghiên cứu tại phòng thí nghiệm Intelligent Communication Networks, trung tâm BK.AI. Nghiên cứu của anh liên quan tới tối ưu, học tăng cường và mạng



trí tuệ nhân tạo, học máy, tối ưu hoá.

Phạm Minh Khiêm hiện tại là sinh viên năm thứ tư tại trường Công nghệ thông tin và Truyền thông, Đại học Bách Khoa Hà Nội. Đồng thời anh cũng đang là trợ lý nghiên cứu tại phòng thí nghiệm Intelligent Communication Networks, trung tâm BK.AI. Nghiên cứu của anh liên quan tới



Đinh Văn Hiếu hiện tại là sinh viên năm thứ tư tại trường Công nghệ thông tin và Truyền thông, Đại học Bách Khoa Hà Nội. Đồng thời anh cũng đang là trợ lý nghiên cứu tại phòng thí nghiệm Intelligent Communication Networks, trung tâm BK.AI. Nghiên cứu của anh liên quan tới mạng IoT, học tăng cường.



Nguyễn Đình Tuấn Anh hiện tại là sinh viên năm thứ năm tại trường Công nghệ thông tin và Truyền thông, Đại học Bách Khoa Hà Nội. Đồng thời anh cũng đang là trợ lý nghiên cứu tại phòng thí nghiệm Intelligent Communication Networks, trung tâm BK.AI. Nghiên cứu của anh liên quan tới mô hình hoá các loại hình mạng, tối ưu hoá, học tăng cường.



TS. Nguyễn Thanh Hùng nhận bằng tiến sỹ khoa học máy tính tại đại học Grenoble, Pháp, năm 2011. Hiện nay, ông là giảng viên tại trường Công nghệ Thông tin và Truyền thông, đại học Bách Khoa Hà Nội. Lĩnh vực nghiên cứu chính của ông là công nghệ phần mềm, trí tuệ nhân tạo, mạng thông minh.



TS. Nguyễn Phi Lê nhận bằng cử nhân và thạc sỹ tại đại học Tokyo, Nhật Bản năm 2007 và 2010. Bà nhận bằng tiến sỹ tin học tại viện Tin học quốc gia Nhật Bản, NII, năm 2016. Hiện nay, bà là giảng viên tại trường Công nghệ Thông tin và Truyền thông, đại học Bách Khoa Hà Nội. TS Lê có hơn 50 công trình công bố tại các hội thảo

và tạp chí uy tín. Bà tham gia phân biện cho nhiều hội thảo và tạp chí uy tín bao gồm ICC, WCNC, Globecom, VTC, IEEE Transactions on Vehicular Technology, IEEE Communications Letters, International Journal of Distributed Sensor Networks, IEICE Transactions on Communications, Journal of Computer Networks and Communications, Wireless Communications and Mobile Computing. Bà nằm trong hội đồng kỹ thuật của nhiều hội nghị khoa học như AI4IoT-2020, SoICT 2019, IEEE ICC'20-CQRM symposium, IEEE ICC'21-CQRM symposium, EAI MONAMI 2020, ICCE 2020.

Các nghiên cứu của bà tập trung vào ứng dụng trí tuệ nhân tạo trong việc tối ưu hoá hoạt động của các mạng thể hệ mới (mạng IoT, MEC, V2X, ...), khoa học dữ liệu và học máy.