THIẾT KẾ HÀM PHẦN THƯỞNG ĐA MỤC TIÊU CHO HỌC TĂNG CƯỜNG TRONG ĐIỆN TOÁN BIÊN HỖ TRỢ ỨNG DỤNG THỰC TẾ ẢO VÀ TĂNG CƯỜNG

Hoàng Trọng Nghĩa[1]
Email: htnghia2@hou.edu.vn, ORCID: 0009-0005-1178-2540

Ngày tòa soạn nhận được bài báo: 10/12/2025. Ngày phản biện đánh giá: 04/06/2026.
Ngày bài báo được duyệt đăng: 23/06/2026.
DOI: 10.59266/houjs.2026.1339

Tóm tắt: *Quản lý động tài nguyên trong hệ thống điện toán biên đa truy cập (Multi-Access Edge Computing - MEC) phục vụ ứng dụng thực tế ảo và thực tế tăng cường (AR/VR) đặt ra nhiều thách thức lớn do yêu cầu khắt khe về độ trễ thấp và hiệu quả năng lượng. Học tăng cường sâu (Deep Reinforcement Learning - DRL) đã trở thành cách tiếp cận phổ biến cho bài toán này; tuy nhiên, hiệu năng của DRL phụ thuộc nhiều vào thiết kế hàm phần thưởng. Mục tiêu của bài báo là phân tích và so sánh ba thiết kế hàm phần thưởng đa mục tiêu cho thuật toán Proximal Policy Optimization (PPO) trong môi trường MEC AR/VR, bao gồm: tổng có trọng số tuyến tính (LWS), hàm logarit (LOG) và phương án phân tầng thích ứng (Adaptive Hierarchical - AH) được nhóm tác giả đề xuất. Phương pháp nghiên cứu là xây dựng môi trường mô phỏng MEC AR/VR tùy chỉnh trên Python với 10 thiết bị người dùng và một trạm gốc tích hợp một máy chủ biên duy nhất, sau đó huấn luyện và đánh giá ba phương án trên bốn chỉ số: độ trễ trung bình, năng lượng tiêu thụ, tỷ lệ vi phạm hạn và tốc độ hội tụ; mỗi cấu hình được lặp lại 5 lần với 5 hạt giống ngẫu nhiên khác nhau. Bài báo cũng bổ sung so sánh với ba baseline đơn giản (Random, Greedy battery-aware, AllLocal) và phân tích độ nhạy của tham số trọng số động κ. Kết quả cho thấy hàm phần thưởng AH đề xuất giảm 19% độ trễ trung bình, 27% năng lượng tiêu thụ và 27% tỷ lệ vi phạm hạn so với LWS, đồng thời cho thấy độ ổn định cao trong dải κ rộng.*

Từ khóa: *điện toán biên, học tăng cường sâu, hàm phần thưởng đa mục tiêu, ứng dụng AR/VR, độ trễ thấp, PPO, quản lý tài nguyên*

I. Đặt vấn đề

Sự phát triển nhanh chóng của các ứng dụng đòi hỏi độ trễ cực thấp như thực tế tăng cường và thực tế ảo (Augmented Reality (AR)/Virtual Reality (VR)), xe tự hành và Internet vạn vật công nghiệp đã đặt ra yêu cầu mới cho hạ tầng mạng thế hệ tiếp theo. Đối với ứng dụng AR/VR, các nghiên cứu chỉ ra rằng độ trễ cần được giữ dưới ngưỡng 30 ms để duy trì trải nghiệm người dùng tự nhiên và tránh hiện tượng say chuyển động (motion sickness) (Siriwardhana & cộng sự, 2021). Yêu cầu này không thể đáp ứng nếu mọi tác vụ

[1] Trường Đại học Mở Hà Nội, Hà Nội, Việt Nam

tính toán đều được đẩy lên đám mây xa do độ trễ truyền dẫn quá lớn. Điện toán biên đa truy cập đã được đề xuất như một kiến trúc khả thi cho phép xử lý tác vụ tại các máy chủ biên đặt gần thiết bị người dùng (Mach & Becvar, 2017; Mao & cộng sự, 2017).

Bài toán cốt lõi trong MEC là quản lý tài nguyên động: với mỗi tác vụ tới, hệ thống phải quyết định xử lý cục bộ tại thiết bị người dùng hay đẩy lên máy chủ biên, đồng thời phân chia băng thông và năng lực xử lý hợp lý. Bài toán này thuộc lớp bài toán quyết định tuần tự dưới sự bất định của môi trường (kênh vô tuyến, hàng đợi, năng lượng pin), thường được hình thức hóa thành quá trình quyết định Markov.

Học tăng cường sâu đã được chứng minh có khả năng học các chính sách phân chia tác vụ thông minh mà không cần mô hình toán học hoàn chỉnh của môi trường (Nguyen & cộng sự, 2019). Một trong những thuật toán DRL phổ biến nhất hiện nay là Proximal Policy Optimization (PPO), một phương pháp tối ưu chính sách cận kề có tính ổn định cao và dễ triển khai (Schulman & cộng sự, 2017). Trong các nghiên cứu trước đây áp dụng DRL cho MEC, hàm phần thưởng thường được thiết kế dưới dạng tổng có trọng số tuyến tính của các mục tiêu thành phần như độ trễ và năng lượng (Chen & cộng sự, 2021; Cao & cộng sự, 2020). Từ đó bộc lộ ba khoảng trống nghiên cứu: (i) hàm phần thưởng thường được kế thừa mặc định mà chưa được khảo sát như một biến số nghiên cứu độc lập, nên ảnh hưởng của nó tới hiệu năng chưa được lượng hóa; (ii) trọng số được cố định trong suốt quá trình huấn luyện, không phản ánh sự thay đổi ưu tiên theo trạng thái pin của thiết bị; (iii) ràng buộc thời hạn được xử lý như một số hạng phạt cộng thêm, khiến tác tử có thể đánh đổi vi phạm thời hạn lấy tiết kiệm năng lượng, điều không chấp nhận được với AR/VR.

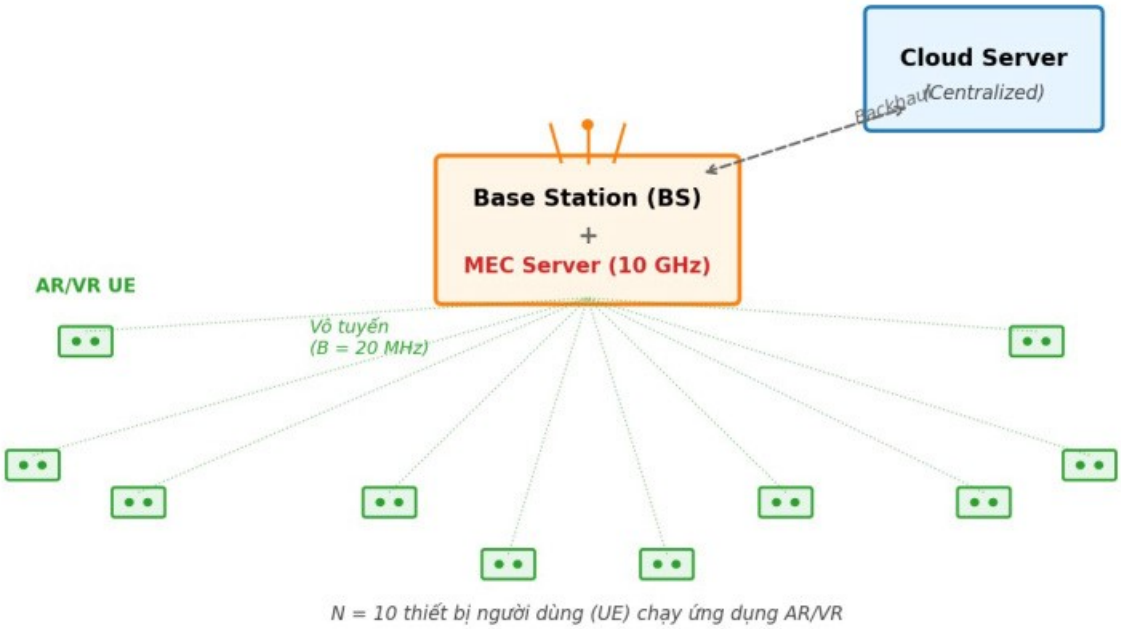
Bài báo này tập trung phân tích và so sánh ba thiết kế hàm phần thưởng đa mục tiêu cho thuật toán PPO trong môi trường MEC AR/VR, gồm: (1) tổng có trọng số tuyến tính (LWS), (2) hàm logarit (LOG), và (3) phân tầng thích ứng AH được nhóm tác giả đề xuất. Đóng góp chính của bài báo gồm: (i) xây dựng môi trường mô phỏng MEC AR/VR tùy chỉnh trên Python phản ánh các tham số kỹ thuật của ứng dụng AR/VR; (ii) đề xuất hàm phần thưởng phân tầng thích ứng kết hợp ràng buộc cứng cho thời hạn với cơ chế trọng số động theo trạng thái pin; (iii) so sánh thực nghiệm ba thiết kế trên bốn chỉ số, mỗi cấu hình được chạy năm lần với năm hạt giống ngẫu nhiên khác nhau; (iv) bổ sung so sánh với ba baseline đơn giản từ tài liệu (Random, Greedy battery-aware, AllLocal) và phân tích độ nhạy của tham số trọng số động κ trong AH.

II. Phương pháp, vật liệu nghiên cứu

2.1. Mô hình hệ thống MEC cho ứng dụng AR/VR

Bài báo xem xét một hệ thống MEC như minh họa trong Hình 1, bao gồm N thiết bị người dùng (UE) chạy ứng dụng AR/VR, một trạm gốc tích hợp một máy chủ biên duy nhất, và một máy chủ đám mây tập trung. Mỗi UE phát sinh các tác vụ tính toán theo chu kỳ tương ứng với việc xử lý một khung hình AR/VR. Tại mỗi bước thời gian t, mỗi UE phải quyết định cách xử lý tác vụ theo ba lựa chọn: thực hiện cục bộ tại UE, đẩy tác vụ tới

máy chủ biên với mức ưu tiên thấp, hoặc đẩy tác vụ với mức ưu tiên cao. Mức ưu tiên cao cho phép tác vụ được xử lý sớm hơn trong hàng đợi tại máy chủ biên nhưng có thể tốn thêm năng lượng truyền. Máy chủ đám mây đóng vai trò dự phòng và không được mô hình hóa chi tiết trong nghiên cứu này.



Hình 1. Mô hình hệ thống MEC cho ứng dụng AR/VR
(1 trạm gốc tích hợp 1 MEC + 1 Cloud)

Tham số hệ thống được thiết lập theo các giá trị tham chiếu phổ biến trong các nghiên cứu MEC: băng thông kênh truyền 20 MHz, công suất phát của UE 100 mW, tần số CPU UE 1 GHz, tần số CPU máy chủ biên 10 GHz, kích thước tác vụ AR/VR từ 50 KB đến 200 KB, số chu kỳ CPU cần thiết từ 30 triệu đến 150 triệu, và thời hạn 30 ms cho mỗi khung hình. Số UE được giữ ở mức N = 10 nhằm tập trung vào phân tích so sánh ba thiết kế hàm phần thưởng trong điều kiện tài nguyên có giới hạn rõ rệt.

2.2. Hình thức hóa bài toán dưới dạng MDP

Bài toán quản lý tài nguyên được hình thức hóa thành quá trình quyết định Markov (MDP) với bộ ngũ <S, A, P, R, γ>, trong đó:

Không gian trạng thái S gồm 5 chiều: (1) độ lợi kênh chuẩn hóa, (2) mức pin còn lại, (3) tải hàng đợi tại máy chủ biên, (4) kích thước tác vụ chuẩn hóa, (5) số chu kỳ CPU cần xử lý chuẩn hóa.

Không gian hành động A có ba lựa chọn rời rạc: a = 0 thực hiện cục bộ; a = 1 đẩy tác vụ với ưu tiên thấp; a = 2 đẩy tác vụ với ưu tiên cao.

Hàm chuyển trạng thái P được sinh trực tiếp từ mô hình mô phỏng (kênh Rayleigh, mô hình Shannon-Hartley, mô hình tiêu thụ năng lượng động). Hệ số chiết khấu γ = 0.99. Hàm phần thưởng R là biến số chính được nghiên cứu trong bài báo này và được chi tiết hóa trong tiểu mục 2.3.

2.3. Ba thiết kế hàm phần thưởng được so sánh

Đặt Lt là độ trễ chuẩn hóa (latency_ ms/THỜI HẠN), Et là năng lượng chuẩn hóa (energy_J/0.05), mt là chỉ thị nhị phân vi phạm thời hạn (mt= 1 nếu latency > thời hạn, 0 nếu ngược lại). Ba hàm phần thưởng được định nghĩa như sau:

Phương án 1 (LWS, Linear Weighted Sum): $Rt = -\alpha \cdot Lt - \beta \cdot Et - \gamma\_miss \cdot m\_t$, với $\alpha = 0.5$, $\beta = 0.3$, $\gamma\_miss = 1.0$. Đây là dạng thiết kế cơ bản được sử dụng phổ biến trong các nghiên cứu DRL cho MEC.

Phương án 2 (LOG, Logarithmic): $Rt = -\alpha \cdot \log(1 + L\_t) - \beta \cdot \log(1 + Et) - \gamma\_miss \cdot m\_t$. Hàm logarit được kỳ vọng làm giảm độ nhạy của tín hiệu phần thưởng với các giá trị cực đoan của Ltvà Et.

Phương án 3 (AH, Adaptive Hierarchical) do tác giả đề xuất: hàm phần thưởng phân tầng với ràng buộc cứng cho thời hạn và trọng số thích ứng theo pin. Cụ thể: nếu mt= 1, $Rt = -2.0 - 0.5 \cdot (Lt - 1)$ (tầng 1: phạt cứng); nếu mt= 0, $Rt = -\alpha \cdot Lt - \beta_{eff} \cdot Et + slack\_bonus$ (tầng 2), trong đó $\beta_{eff} = \beta + \kappa \cdot (1 - battery)$ là trọng số năng lượng động phụ thuộc trạng thái pin ($\kappa = 0.4$ mặc định, được phân tích độ nhạy ở Mục 3.4), và $slack\_bonus = 0.5 \cdot \max(0, 1 - L\_t)$ khi Lt< 0.5 nhằm khuyến khích các hành động hoàn thành sớm.

Sự khác biệt cốt lõi của AH so với LWS và LOG là: thay vì xử lý mt như một thành phần cộng thêm với trọng số cố định, AH coi mt là ràng buộc cứng và áp dụng tín hiệu phần thưởng riêng biệt cho hai chế độ: tránh vi phạm và tối ưu mềm.

2.4. Cấu hình mô phỏng và đánh giá

Mô phỏng được triển khai trên ngôn ngữ Python với thư viện PyTorch cho phần mạng neural và NumPy cho phần môi trường MEC. Thuật toán PPO sử dụng kiến trúc Actor-Critic với hai lớp ẩn 64 đơn vị và hàm kích hoạt Tanh, learning rate $3 \cdot 10^{-4}$, Kepochs = 4 cho mỗi lần cập nhật, hệ số cắt ratio $\varepsilon = 0.2$, hệ số giá trị 0.5 và hệ số entropy 0.01. Mỗi episode dài 50 bước thời gian. Mỗi cấu hình hàm phần thưởng được huấn luyện 200 episode và lặp lại 5 lần với 5 hạt giống ngẫu nhiên khác nhau (42, 7, 123, 2024, 88) để tính giá trị trung bình và độ lệch chuẩn.

Bốn chỉ số đánh giá được sử dụng: (1) độ trễ trung bình tính bằng mili-giây, (2) năng lượng tiêu thụ tính bằng mili-joule, (3) tỷ lệ vi phạm hạn tính bằng phần trăm, và (4) tốc độ hội tụ tính bằng số episode cần thiết để đạt 90% mức phần thưởng cuối.

Để đánh giá khách quan đóng góp của thuật toán DRL so với các chiến lược đơn giản từ tài liệu, bài báo bổ sung ba chính sách baseline không học: (i) Random: Chọn ngẫu nhiên một trong ba hành động tại mỗi bước; (ii) Greedy battery-aware: Heuristic cổ điển trước thời DRL: nếu pin > 0.5 thì xử lý cục bộ, ngược lại đẩy với ưu tiên cao (Mach & Becvar, 2017); (iii) AllLocal: Luôn xử lý cục bộ tại UE, đóng vai trò cận dưới khi không có MEC. Ba baseline này được chạy với cùng năm hạt giống ngẫu nhiên để bảo đảm so sánh công bằng.

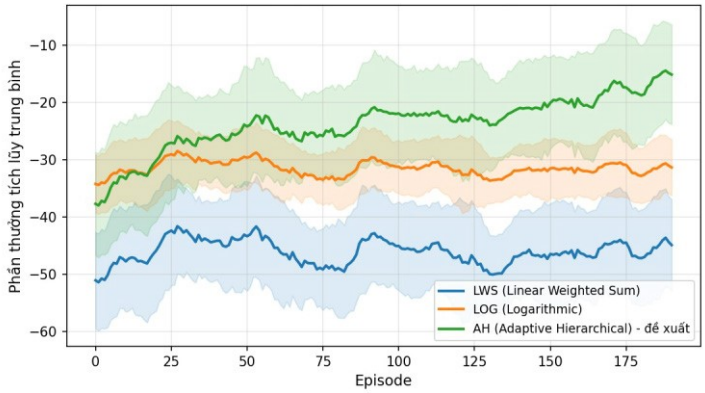
Về độ phức tạp tính toán, mạng Actor-Critic với hai lớp ẩn 64 đơn vị cần khoảng 4.6 nghìn phép nhân cộng cho mỗi lần ra quyết định và dưới 5 nghìn tham số, không đáng kể so với ngân sách độ trễ 30 ms của một khung hình AR/VR. Ba thiết kế hàm phần thưởng dùng chung kiến trúc mạng và số vòng lặp huấn luyện nên có

chi phí tính toán tương đương; khác biệt giữa chúng chỉ là vài phép toán vô hướng khi tính giá trị phần thưởng.

III. Kết quả và thảo luận

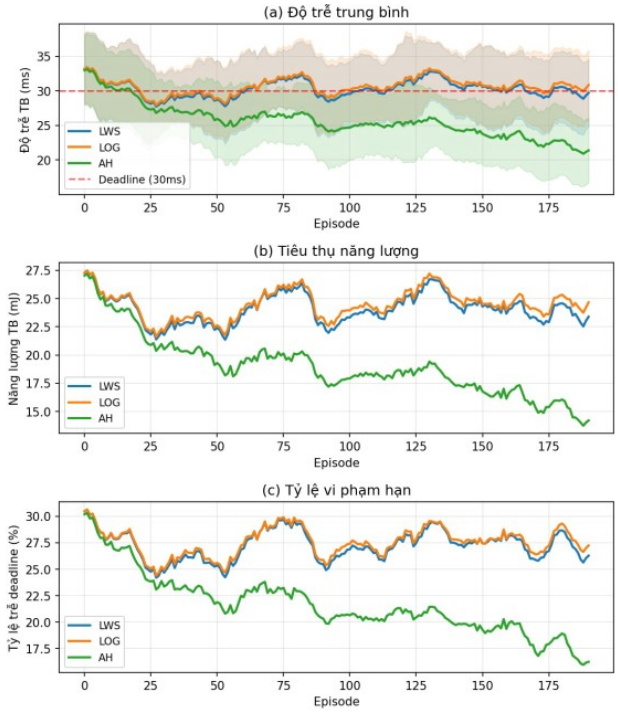
3.1. Đường cong hội tụ

Hình 2 mô tả đường cong phần thưởng tích lũy của ba phương án theo số episode huấn luyện. Do thang đo phần thưởng khác nhau giữa ba phương án, ta chỉ so sánh hình dạng và xu hướng của đường cong. Đường cong AH có xu hướng tăng đều và ổn định trong toàn bộ quá trình huấn luyện, cho thấy tác tử tiếp tục cải thiện chính sách. Trong khi đó, đường cong LWS dao động quanh một mức cố định ngay từ giai đoạn đầu, gợi ý rằng tác tử đã rơi vào nghiệm tối ưu cục bộ. Đường cong LOG có hành vi tương tự nhưng với độ dao động nhỏ hơn do tính chất nén thang đo của hàm logarit.



Hình 2. Đường cong hội tụ của ba thiết kế hàm phần thưởng (trung bình ± độ lệch chuẩn trên 5 hạt giống)

3.2. Hiệu năng theo bốn chỉ số đánh giá



Hình 3. Diễn biến độ trễ, năng lượng và tỷ lệ vi phạm hạn theo episode

Hình 3 trình bày diễn biến của ba chỉ số chính gồm độ trễ, năng lượng và tỷ lệ vi phạm hạn theo thời gian huấn luyện. Quan sát hình 3(a), AH duy trì độ trễ trung bình thấp hơn rõ rệt và giảm dần xuống dưới ngưỡng thời hạn 30 ms, trong khi LWS và LOG dao động quanh chính ngưỡng này. Tương tự, hình 3(b) cho thấy AH liên tục giảm năng lượng tiêu thụ trong quá trình huấn luyện, trong khi LWS và LOG gần như đứng yên. Hình 3(c) cho thấy tỷ lệ vi phạm hạn của AH cải thiện liên tục, còn hai phương án kia không cải thiện đáng kể.

Bảng 1 tổng hợp giá trị trung bình của các chỉ số trên 50 episode cuối (sau khi đã hội tụ) cùng với độ lệch chuẩn tính trên 5 hạt giống ngẫu nhiên.

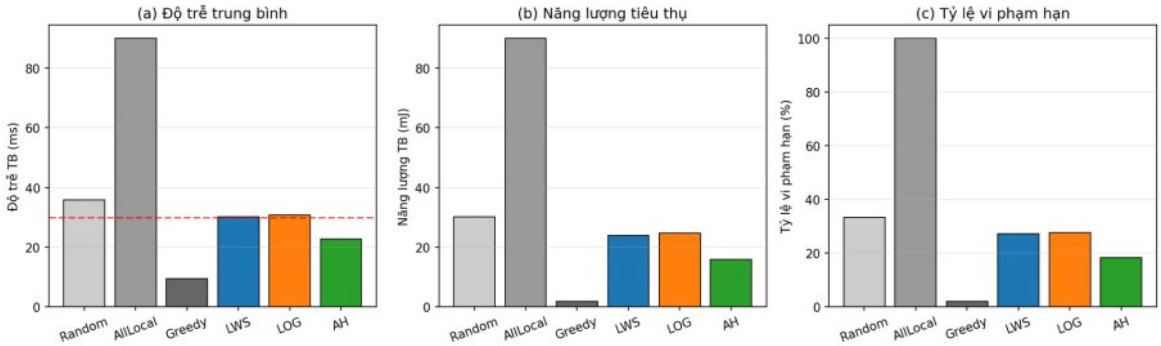
Bảng 1. So sánh hiệu năng cuối của ba thiết kế hàm phần thưởng (trung bình ± std, 5 seeds)

Phương án	Độ trễ (ms)	Năng lượng (mJ)	Tỷ lệ vi phạm (%)
LWS	30.13 ± 2.70	23.92 ± 2.91	27.29 ± 2.82
LOG	30.73 ± 2.64	24.59 ± 2.84	27.74 ± 2.70
AH (đề xuất)	**24.35 ± 2.88**	**17.56 ± 3.18**	**20.07 ± 3.64**

Kết quả từ Bảng 1 cho thấy AH cải thiện rõ rệt cả ba chỉ số so với LWS và LOG, tương ứng 19.2% về độ trễ, 26.6% về năng lượng tiêu thụ và 26.5% về tỷ lệ vi phạm hạn. Kiểm định t Welch hai phía trên 5 hạt giống xác nhận toàn bộ sáu cặp so sánh giữa AH với LWS và AH với LOG đều có ý nghĩa thống kê ($p < 0.05$), với hệ số quy mô hiệu ứng Cohen's d từ 2.07 đến 2.39, tương ứng mức hiệu ứng rất lớn.

3.3. So sánh với baseline tham chiếu từ tài liệu

Để định vị hiệu năng của các phương án DRL trong bối cảnh rộng hơn, bài báo so sánh ba thiết kế hàm phần thưởng với ba baseline đơn giản đã giới thiệu ở Mục 2.4. Hình 4 trình bày so sánh hiệu năng cuối giữa ba phương án PPO và ba baseline.



Hình 4. So sánh hiệu năng trung bình tại trạng thái đã hội tụ của ba thiết kế hàm phần thưởng và ba baseline

Bảng 2 trình bày số liệu chi tiết của các baseline.

Bảng 2. Hiệu năng của ba baseline đơn giản (trung bình ± std, 5 seeds)

Baseline	Logic	Độ trễ (ms)	Vi phạm (%)
Random	Chọn ngẫu nhiên một trong ba hành động	35.72 ± 0.75	33.29 ± 0.77
Greedy	Pin > 0.5 → cục bộ; ngược lại → ưu tiên cao	9.49 ± 0.03	2.00 ± 0.00
AllLocal	Luôn xử lý cục bộ (không có MEC)	89.93 ± 0.23	100.00 ± 0.00

Quan sát Bảng 2 và Hình 4, AH vượt trội rõ rệt so với AllLocal (vi phạm hạn 100% do tài nguyên cục bộ không đủ cho tác vụ AR/VR) và Random (35.72 ms / 33.29% vi phạm). Đáng chú ý, baseline Greedy battery-aware cho hiệu năng rất tốt (9.49 ms / 2.00% vi phạm), thậm chí tốt hơn AH ở các chỉ số tuyệt đối. Nguyên nhân là trong môi trường mô phỏng đơn giản, một heuristic thủ công nắm rõ cấu trúc bài toán có thể đạt hiệu năng cao, còn lợi thế của DRL chỉ bộc lộ trong môi trường phức tạp hơn với nhiều tác tử, nhiều TRẠM GỐC và tải động (Nguyen & cộng sự, 2019).

Bảng 3 trình bày so sánh tham chiếu với hai nghiên cứu DRL gần đây áp dụng cho MEC. Do cấu hình hệ thống của mỗi nghiên cứu khác nhau, các con số dưới đây chỉ nhằm định vị hiệu năng của AH, không phải so sánh trực tiếp.

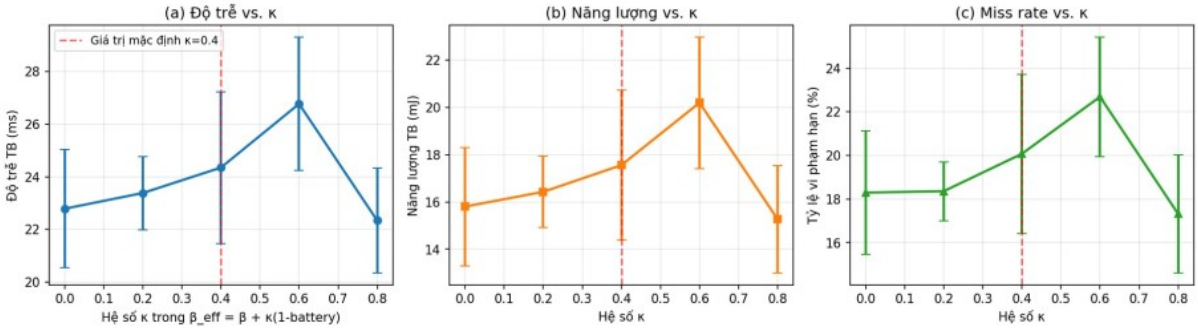
Bảng 3. Tham chiếu kết quả từ một số nghiên cứu DRL/MADRL gần đây cho MEC

Nghiên cứu	Cấu hình	Độ trễ	Năng lượng
Chen và cộng sự (2021), DRL-DQN	5 UE, 1 server, MEC tổng quát	25-40 ms (tùy task)	-
Cao và cộng sự (2020), MADRL-DQN	6 UE, multi-channel + Internet vạn vật công nghiệp	-	giảm 30-40% so heuristic
AH (đề xuất)	10 UE, 1 TRẠM GỐC + 1 MEC, AR/VR	24.35 ms	17.56 mJ

Bảng 3 cho thấy độ trễ trung bình của AH (24.35 ms) nằm ở cận dưới dải mà Chen và cộng sự (2021) báo cáo (25-40 ms), còn mức giảm năng lượng so với LWS (26.6%) tiệm cận mức độ lợi mà Cao và cộng sự (2020) công bố cho MADRL-DQN. Như vậy AH cho hiệu năng nằm trong dải cạnh tranh của các phương pháp DRL gần đây.

3.4. Phân tích độ nhạy của tham số trọng số động κ

Phương án AH chứa tham số κ trong công thức $\beta_{eff} = \beta + \kappa \cdot (1 - \text{battery})$, điều khiển mức độ phụ thuộc của trọng số năng lượng vào trạng thái pin. Để đánh giá ảnh hưởng của κ đến hiệu năng tổng thể, bài báo thực hiện phân tích độ nhạy bằng cách quét κ trong tập {0.0, 0.2, 0.4, 0.6, 0.8}. Mỗi giá trị κ được chạy với 5 hạt giống ngẫu nhiên độc lập, trong đó hạt giống được dịch chuyển theo κ ($sd_{eff} = sd + [1000 \cdot \kappa]$) để bảo đảm mỗi giá trị κ có 5 quỹ đạo môi trường ngẫu nhiên riêng biệt.



Hình 5. Phân tích độ nhạy của tham số trọng số động κ trong AH, 5 hạt giống mỗi giá trị eff

Bảng 4 tổng hợp số liệu phân tích độ nhạy.

Bảng 4. Hiệu năng AH theo các giá trị tham số κ (trung bình ± std, 5 hạt giống)

κ	Độ trễ (ms)	Năng lượng (mJ)	Vi phạm (%)
0.0	22.79 ± 2.25	15.81 ± 2.50	18.30 ± 2.83
0.2	23.38 ± 1.39	16.43 ± 1.51	18.36 ± 1.35
0.4 (mặc định)	24.35 ± 2.88	17.56 ± 3.18	20.07 ± 3.64
0.6	26.78 ± 2.54	20.20 ± 2.78	22.69 ± 2.74
0.8	22.34 ± 2.00	15.27 ± 2.27	17.32 ± 2.71

Hình 5 và Bảng 4 có hai nhận xét:

- Hiệu năng của AH biến động trong một dải hẹp với mọi $\kappa \in [0, 0.8]$, và mọi giá trị κ đều vượt trội LWS với mức cải thiện độ trễ tối thiểu 11% ngay ở trường hợp xấu nhất $\kappa = 0.6$. Điều này chứng tỏ AH ổn định cao đối với việc lựa chọn κ, một ưu điểm thực tiễn vì giảm gánh nặng tinh chỉnh tham số khi triển khai.

- Sự khác biệt giữa các giá trị κ là có ý nghĩa thống kê: khoảng tin cậy 95% của $\kappa = 0.6$ ([24.55, 29.01] ms) không chồng lấn với khoảng của $\kappa = 0$ và $\kappa = 0.8$, cho thấy $\kappa = 0.6$ thực sự kém hơn hai đầu mút của dải khảo sát.

Cấu hình mặc định $\kappa = 0.4$ được chọn làm thỏa hiệp giữa độ nhạy theo pin của β_{eff} và sự ổn định của tín hiệu phần thưởng; việc tinh chỉnh κ theo từng kịch bản tải là hướng có thể khai thác thêm.

3.5. Phân tích nguyên nhân và đánh đổi

Phương án LWS hội tụ rất nhanh nhưng đạt nghiệm chất lượng kém. Lý do là tín hiệu phần thưởng có cấu trúc tuyến tính khiến tác tử dễ rơi vào nghiệm tối ưu cục bộ ưu tiên giảm năng lượng nhưng không tránh được vi phạm thời hạn. Phương án LOG có cùng đặc điểm hội tụ nhanh nhưng nghiệm thậm chí hơi tệ hơn LWS, do hàm logarit nén thang đo dẫn đến tín hiệu phần thưởng quá yếu để định hướng cập nhật chính sách hiệu quả.

Phương án AH cần nhiều episode hơn để hội tụ, nhưng đây là đánh đổi xứng đáng. Ràng buộc cứng tại tầng 1 buộc tác tử ưu tiên tránh vi phạm thời hạn trước, sau đó mới cân bằng độ trễ và năng lượng tại tầng 2. Cơ chế trọng số thích ứng theo pin (β_{eff} tăng khi pin giảm) giúp tác tử tiết kiệm năng lượng khi pin yếu, còn cơ chế thưởng hoàn thành sớm khuyến khích hoàn thành sớm thay vì chỉ vừa đủ kịp thời hạn. Ba cơ chế này tạo nên tín hiệu phần thưởng giàu thông tin hơn, dẫn hướng tìm kiếm chính sách tới nghiệm có chất lượng cao hơn.

Tuy nhiên tỷ lệ vi phạm hạn của AH (20.07%) vẫn cao hơn mức mong muốn cho ứng dụng AR/VR thực tế. Điều này cho thấy giới hạn vốn có của hệ thống trong cấu hình tài nguyên đã chọn: với 10 UE chia sẻ một máy chủ biên 10 GHz và băng thông 20 MHz, một số tác vụ vẫn không thể đáp ứng thời hạn 30 ms trong điều kiện kênh xấu. Mở rộng sang kiến trúc đa tác tử với nhiều TRẠM GỐC, kết hợp cơ chế dự đoán kênh, là hướng cải thiện quan trọng trong tương lai.

IV. Kết luận

Bài báo đã đề xuất và đánh giá thiết kế hàm phần thưởng phân tầng thích ứng cho thuật toán PPO trong môi trường điện

toán biên hỗ trợ ứng dụng AR/VR. Hàm phần thưởng đề xuất tổ chức tín hiệu thưởng phạt thành hai tầng: ràng buộc cứng cho thời hạn và tối ưu mềm với trọng số thích ứng theo trạng thái pin. Trên môi trường mô phỏng 10 UE và một máy chủ biên, AH giảm 19% độ trễ, 27% năng lượng và 27% tỷ lệ vi phạm hạn so với hàm phần thưởng tuyến tính, khác biệt được xác nhận bằng kiểm định t Welch ($p < 0.05$).

So sánh với ba baseline đơn giản (Random, Greedy battery-aware, AllLocal) cho thấy AH thắng rõ rệt Random và AllLocal, nhưng heuristic Greedy đơn giản vẫn cho hiệu năng tốt hơn ở chỉ số tuyệt đối. Điều này gợi ý rằng giá trị thực sự của các khung DRL nằm ở tính tổng quát và khả năng mở rộng sang các kịch bản phức tạp hơn nơi heuristic thủ công khó áp dụng. Phân tích độ nhạy với tham số trọng số động κ cho thấy AH có tính ổn định cao trong dải κ rộng, với mọi giá trị $\kappa \in [0, 0.8]$ đều vượt trội LWS ít nhất 11% về độ trễ.

Đóng góp khoa học của nghiên cứu là bằng chứng thực nghiệm có cơ sở thống kê về vai trò của thiết kế hàm phần thưởng trong DRL cho MEC, kèm một mẫu hàm phần thưởng có thể tái sử dụng. Về thực tiễn, kỹ thuật đề xuất áp dụng được cho các hệ thống MEC hỗ trợ AR/VR và các ứng dụng truyền thông độ trễ thấp siêu tin cậy như xe tự hành hay Internet vạn vật công nghiệp.

Hướng nghiên cứu tiếp theo bao gồm: (i) mở rộng sang môi trường đa tác tử với nhiều UE và nhiều TRẠM GỐC hợp tác hoặc cạnh tranh (hướng mà nhóm tác giả đã bước đầu khai thác trong Hoàng (2026)); (ii) tích hợp cơ chế chú ý để xử lý số lượng tác tử thay đổi động; (iii) đánh giá trên môi trường mô phỏng tiêu chuẩn ngành như ns-3 hoặc thử nghiệm trên thiết bị thực USRP; (iv) nghiên cứu cơ chế tinh chỉnh κ thích nghi theo loại tải để khai thác triệt để dải hiệu năng tốt nhất của AH.

Tài liệu tham khảo

Cao, Z., Zhou, P., Li, R., Huang, S., & Wu, D. (2020). Multiagent Deep Reinforcement Learning for Joint Multichannel Access and Task Offloading of Mobile-Edge Computing in Industry 4.0. *IEEE Internet of Things Journal, 7*(7), 6201-6213. https://doi.org/10.1109/JIOT.2020.2968951

Chen, J., Xing, H., Xiao, Z., Xu, L., & Tao, T. (2021). A DRL Agent for Jointly Optimizing Computation Offloading and Resource Allocation in MEC. *IEEE Internet of Things Journal, 8*(24), 17508-17524. https://doi.org/10.1109/JIOT.2021.3081694

Hoàng, T. N. (2026). Đề xuất cơ chế truyền thông trạng thái thích nghi dựa trên mức độ quan trọng cho bài toán phân tải tác vụ đa tác tử trong điện toán biên di động. *Tạp chí Khoa học Trường Đại học Mở Hà Nội*, *số đặc biệt 10A*, 25-35. https://doi.org/10.59266/houjs.2026.1151

Mach, P., & Becvar, Z. (2017). Mobile edge computing: A Survey on Architecture and Computation Offloading. *IEEE Communications Surveys & Tutorials, 19*(3), 1628-1656. https://doi.org/10.1109/COMST.2017.2682318

Mao, Y., You, C., Zhang, J., Huang, K., & Letaief, K. B. (2017). A Survey on Mobile Edge Computing: The Communication Perspective. *IEEE Communications Surveys & Tutorials, 19*(4), 2322-2358. https://doi.org/10.1109/COMST.2017.2745201

Nguyen, C. L., Dinh, T. H., Gong, S., Niyato, D., Wang, P., Liang, Y.-C., & Kim, D. I. (2019). Applications of deep reinforcement learning in

communications and networking: A survey. *IEEE Communications Surveys & Tutorials, 21*(4), 3133-3174. https://doi.org/10.1109/COMST.2019.2916583

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv*. https://doi.org/10.48550/arXiv.1707.06347

Siriwardhana, Y., Porambage, P., Liyanage, M., & Ylianttila, M. (2021). A Survey on Mobile Augmented Reality With 5G Mobile Edge Computing: Architectures, Applications, and Technical Aspects. *IEEE Communications Surveys & Tutorials, 23*(2), 1160-1192. https://doi.org/10.1109/COMST.2021.3061981

DESIGNING A MULTI-OBJECTIVE REWARD FUNCTION FOR REINFORCEMENT LEARNING IN EDGE COMPUTING SUPPORTING AR/VR APPLICATIONS

Hoang Trong Nghia[1]

Abstract: *Dynamic resource management in Multi-Access Edge Computing (MEC) for Augmented and Virtual Reality (AR/VR) applications poses sutrạm gốctantial challenges due to strict requirements on low latency and energy efficiency. Deep Reinforcement Learning (DRL) has become a popular approach for this problem; however, its performance heavily depends on reward function design. This paper analyzes and compares three multi-objective reward function designs for the Proximal Policy Optimization (PPO) algorithm in an MEC AR/VR environment: linear weighted sum (LWS), logarithmic (LOG), and the proposed adaptive hierarchical formulation (AH). The research method involves constructing a custom Python simulation environment with ten user devices and one base station integrated with a single edge server, then training and evaluating the three designs over four metrics, namely average latency, energy consumption, thời hạn miss rate, and convergence speed, with each configuration run using five different random seeds. The paper additionally compares ađộ lợist three simple baselines from the literature (Random, Greedy battery-aware, AllLocal) and conducts a sensitivity analysis of the dynamic weight parameter κ. The results show that the proposed AH reward function reduces average latency by 19%, energy consumption by 27%, and thời hạn miss rate by 27% compared to LWS (Welch t-test, $p < 0.05$), while maintaining high stability across a wide κ range.*

Keywords: *edge computing, deep reinforcement learning, multi-objective reward function, AR/VR applications, low latency, PPO, resource management*

[1] Hanoi Open University, Hanoi, Vietnam