HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

THẠCH QUỐC TUẤN

NÂNG CAO CHẤT LƯỢNG PHÁT VIDEO QUA HTTP BẰNG PHƯƠNG PHÁP HỌC TĂNG CƯỜNG

Chuyên ngành: HỆ THỐNG THÔNG TIN

Mã số: 8.48.01.04

TÓM TẮT LUẬN VĂN THẠC SĨ

(Theo định hướng ứng dụng)

TP. HỒ CHÍ MINH – NĂM 2022

Luận văn được hoàn thành tại:

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Người hướng dẫn khoa học: **PGS.TS VÕ THỊ LƯU PHƯƠNG**

Phản biện 1: .....

Phản biện 2: .....

Luận văn sẽ được bảo vệ trước Hội đồng chấm luận văn tại Học viện Công nghệ Bưu chính Viễn Thông

Vào lúc: ..... giờ ..... ngày ..... tháng ..... năm .....

Có thể tìm hiểu luận văn tại:

- Thư viện của Học viện Công nghệ Bưu Chính Viễn Thông.

MỞ ĐẦU

Với xu hướng phát triển của điện toán đám mây và kết nối vạn vật IoT, thập kỷ vừa qua đã chứng kiến sự phát triển vượt bậc của phát video trực tuyến và chiếm phần lớn lưu lượng truy cập Internet hiện nay nhờ những tiến bộ trong công nghệ truyền tải, năng lực thiết bị đầu cuối và các phương pháp nén âm thanh-video và chiếm hơn 60% lưu lượng Internet toàn cầu [1] , [2]. Thị trường phát video trực tuyến được định giá lên đến hàng tỉ đô la. Cùng với sự phát triển của thị trường này là yêu cầu ngày càng cao các video có chất lượng, đã được chứng minh là một trong những yếu tố quan trọng ảnh hưởng đến trải nghiệm chất lượng của người dùng [3], [4]. Điều này tạo ra những thách thức cho việc cung cấp các video với "Chất lượng trải nghiệm tốt nhất" qua mạng Internet, hệ thống mạng ban đầu được thiết kế để theo kiểu "nỗ lực tối đa" – để truyền tải các dữ liệu không theo thời gian thực. Người dùng có thể dừng xem nếu có các vấn đề với việc phát trực tuyến như chất lượng video thấp hay việc đứng hình, phát lại. Ảnh hưởng trực tiếp đến doanh thu của các nhà cung cấp nội dung video.

Với mục tiêu chính là nâng cao chất lượng trải nghiệm của người dùng, vốn bị ảnh hưởng bởi nhiều yếu tố như băng thông, cường độ tín hiệu, độ nghẽn mạng và thời gian mạng hội tụ sau khi có sự thay đổi, nhiều thuật toán tương thích tốc độ bit [5] được triển khai rộng rãi phía đầu cuối khách hàng và các yêu cầu về mức chất lượng khác nhau đối với máy chủ. Trong những năm gần đây, giải pháp Học tăng cường [6], [7] đang nổi trội và thay thế cho các phương pháp truyền thống khác. Giải pháp end-to-end này học cách cải thiện chất lượng các phiên phát trực tuyến bằng cách sử dụng các tham số đầu vào như là chất lượng mạng và kích thước video, với cách thức tính toán đơn giản hơn. Từ những điều trên, tôi chọn đề tài "**Nâng cao chất lượng phát video qua HTTP bằng phương pháp học tăng cường",** trên cơ sở dựa trên các nghiên cứu trước đó, xây dựng thuật toán ABR dưới hình thức học tăng cường trong môi trường mô phỏng, sử dụng video thời gian thực và mạng 4G. Sau đó, hiệu suất của các thuật toán được đánh giá theo các giao thức đánh giá đã biết. Cuối cùng, xin đề xuất một số hướng nghiên cứu trong tương lai về vấn đề này, cải thiện một số thông số ảnh hưởng đến QoE người dùng.

Luận văn gồm 5 chương chính với các nội dung sau:

Chương 1: Giới thiệu tổng quan về kỹ thuật phát video qua HTTP, hiện trạng phát video trực tuyến hiện nay, vai trò của QoE trong phát video cũng như các yếu tố ảnh hưởng đến QoE.

Chương 2: Trình bày các công trình nghiên cứu có liên quan về các thuật toán tương thích tốc độ bit của phát trực tuyến video, các đánh giá QoE và xây dựng hàm QoE.

Chương 3: Giới thiệu về giải pháp nâng cao chất lượng phát trực tuyến video bằng phương pháp học tăng cường (reinforcement learning). Đề xuất thuật toán học tăng cường sâu Deep Q-Learning (DQN)

Chương 4: Trình bày chi tiết các công cụ mô phỏng sẽ được sử dụng, cùng với bộ dữ liệu đến quá trình mô phỏng và đánh giá kết quả của toàn bộ quá trình.

Chương 5: Kết luận nội dung đã được trong đề tài, nêu những khó khăn, hạn chế trong quá trình nghiên cứu đã gặp phải và đề xuất hướng phát triển tiếp theo.

Đề tài: NÂNG CAO CHẤT LƯỢNG PHÁT VIDEO QUA HTTP BẰNG PHƯƠNG PHÁP HỌC TĂNG CƯỜNG

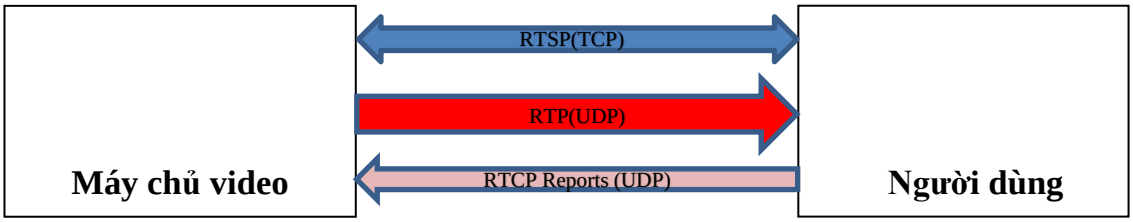
Tóm tắt luận văn

CHƯƠNG 1. TỔNG QUAN VỀ VIDEO STREAMING

1.1. Đặt vấn đề

1.1.1. *1.1.1 Truyền phát video hiện nay*

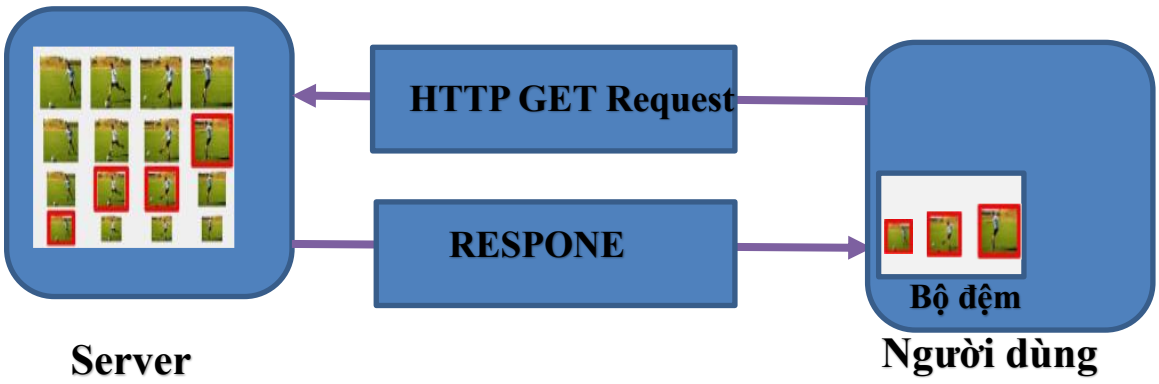
Video là một loại dữ liệu đa phương tiện quan trọng trong lĩnh vực truyền thông và giải trí. Lưu lượng truy cập video tăng trưởng rất nhanh chóng trong thời gian gần đây, và dự kiến chiếm phần lớn lưu lượng Internet toàn cầu [1]. Vào thời kỳ đầu, video được phát với công nghệ chuyển mạch gói, dù sau đó được chuyển qua mạng Internet, vẫn gặp những yếu tố bất lợi về băng thông, độ trễ, và mất gói tin. Phát video qua HTTP [1] là một công nghệ rất phổ biến mà nội dung đa phương tiện được phân phối liên tục từ các máy chủ HTTP đến các thiết bị đầu cuối người dùng.



Hình 1.1: Mô hình phát trực tuyến truyền thống

Trong mô hình phát trực tuyến truyền thống không sử dụng HAS như Hình 1.1, người dùng sẽ nhận được các thông tin đa phương tiện được phát đi từ các máy chủ bằng cách sử dụng các giao thức có thiên hướng kết nối như Real-time Messaging Protocol (RTMP/TCP) hoặc không kết nối như Real-time Transport Protocol (RTP/UDP). Giao thức chung để điều khiển các máy chủ kiểu truyền thống chứa các file nội dung đa phương tiện là giao thức RSTP

(Real-time Streaming Protocol: Giao thức phát trực tuyến thời gian thực). RTSP sẽ chịu trách nhiệm thiết lập phiên trực tuyến và luôn giữ trạng thái kết nối, nhưng nó không chịu trách nhiệm cho việc phân phối thật sự, mà nhiệm vụ phân phối là do RTP. Dựa trên các RTCP Reports (RTP Control Protocol: giao thức điều khiển RTP) từ người dùng, máy chủ có thể thay đổi tốc độ tương thích và lịch trình chuyển phát dữ liệu. Những điều này làm cho máy chủ có cấu trúc phức tạp hơn và đắt đỏ hơn. Hơn nữa, các giao thức hoặc các cấu hình cần được thiết lập xuyên suốt phiên, ngoài ra các luồng dữ liệu đa phương tiện có thể bị chặn lại trong trường hợp sử dụng các thiết bị NAT hoặc tường lửa. Mặc dù triển khai theo các giao thức cơ bản như nhau, nhưng đối với các nhà cung cấp dịch vụ khác nhau, các máy chủ có thể khác nhau về cầu hình hoặc cách vận hành, khi các máy chủ có lỗi sẽ làm cho phiên trực tuyến bị gián đoạn hoặc không được liên tục trừ khi có giải pháp sử dụng máy chủ dự phòng. Những vấn đề như việc phụ thuộc vào nhà cung cấp, khả năng mở rộng và cũng như chi phí bảo trì cao sẽ gây ra những thách thức cho các giao thức như RTSP.



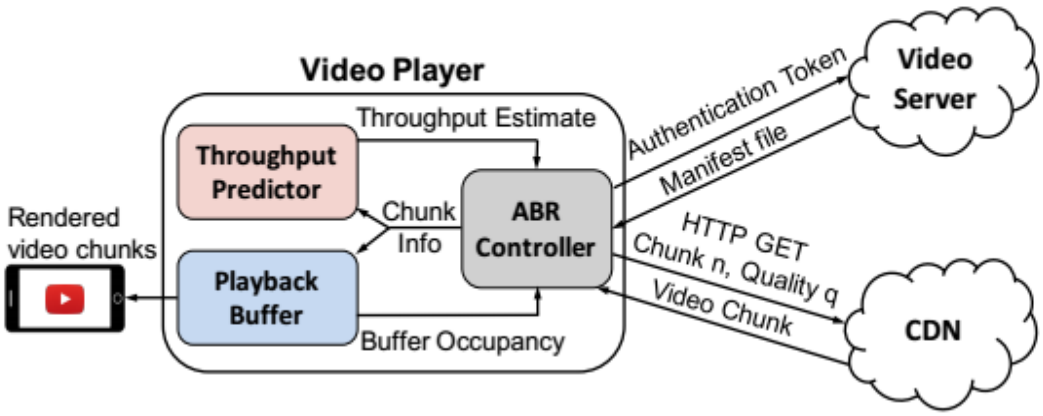
Hình 1.2: Mô hình phát trực tuyến HAS

So với mô hình phát trực tuyến truyền thống, mô hình HAS sử dụng HTTP như là một ứng dụng và sử dụng TCP là giao thức cho lớp truyền tải, và người dùng lấy dữ liệu từ máy chủ HTTP chuẩn như Hình 1.2. Cơ bản, các máy chủ này chỉ chứa nội dung đa phương tiện. Giải pháp HAS triển khai theo cơ chế tương thích động tùy theo nhiều điều kiện kết nối mạng khác nhau để cung cấp trải nghiệm phát trực tuyến liên tục, chí ít cũng mượt mà hơn. File đa

phương tiện như video hoặc luồng dữ liệu phát trực tuyến nhận từ nguồn phát, trước khi được phát sẽ được chuẩn hóa tại máy chủ HTTP. Các file này sẽ được chia nhỏ thành các phân đoạn (còn gọi là chunk) với mức thời lượng tương ứng. Các phân đoạn được mã hóa với các mức tốc độ bit khác nhau, tương ứng với chất lượng khác nhau, bằng cách sử dụng các bộ mã hóa hoặc chuyển mã. Theo đó, máy chủ tạo các file đầu mục, đây là danh sách bao gồm địa chỉ web máy chủ HTTP, các phân đoạn video khả dụng để xác định các phân đoạn thuộc máy chủ nào và thời gian khả dụng. Trong suốt một phiên HAS, đầu tiên người dùng sẽ nhận bảng kê chi tiết bao gồm dữ liệu của video, âm thanh, phụ đề và các tham số khác, sau đó sẽ tiến hành thường xuyên đo đạc các tham số bắt buộc như: băng thông mạng khả dụng, trạng thái bộ đệm, pin và tình trạng CPU, v.v. Người dùng đầu cuối sẽ lựa chọn chất lượng các phân đoạn sẽ được tải xuống tiếp theo trong số các các phân đoạn được lưu trữ tại máy chủ tùy theo các thông số đo đạc được.

Truyền phát video qua HTTP có một số lợi ích là cơ sở hạ tầng Internet đã phát triển để hỗ trợ HTTP một cách hiệu quả. Ngoài ra, hầu hết tất cả các tường lửa đều được cấu hình để hỗ trợ các kết nối của HTTP. Thêm vào đó, với phát trực tuyến qua HTTP, đầu cuối người dùng sẽ quản lý việc truyền phát mà không cần duy trì trạng thái phiên kết nối trên máy chủ. Do đó, việc triển khai dịch vụ với số lượng lớn người dùng không gây tốn kém tài nguyên máy chủ nên hiện nay chủ yếu sử dụng các giao thức hoạt động trên nền tảng HTTP để cung cấp các dịch vụ phát trực tuyến video.

.



Hình 1.3: Tổng quan phát trực tuyến tương thích tốc độ bit qua HTTP

Theo đó, video được lưu trữ tại các máy chủ video, chia thành nhiều phân đoạn, thường là vài giây. Mỗi phân đoạn được mã hóa thành nhiều mức tốc độ bit khác nhau. Phân đoạn có mức tốc độ bit cao hơn đồng nghĩa với chất lượng cao hơn và có kích thước lớn hơn. Mức tốc độ bit của các phân đoạn video được cân chỉnh để truyền phát được mượt mà, liên tục, nghĩa là, các chương trình phát video tại người dùng có thể chuyển sang các mức tốc độ bit khác nhau của các phân đoạn video mà không tác động đến các đoạn dự phòng hoặc không bỏ qua các phần của video.

Hình 1.3 mô tả tiến trình phát video trực tuyến qua HTTP hiện nay.

- Dữ liệu video được chia nhỏ thành các chunk – các phân đoạn video, được mã hóa với các mức chất lượng khác nhau và lưu trữ tại máy chủ (streaming server).
- Phần mềm tại phía người dùng (media player, web browser, ...) cần kết nối đến máy chủ và xác định tệp video trên máy chủ streaming muốn xem.
- Nhà cung cấp dịch vụ sẽ gửi lại cho người dùng danh sách các máy chủ chứa video và danh sách tốc độ bit của các video sẵn có
- Người dùng sẽ yêu cầu từng phân đoạn video, bằng cách sử dụng các thuật toán tương thích tốc độ bit (ABR: Adaptive Bitrate Algorithm). Các thuật toán này sử dụng nhiều thông số đầu vào (như là tình trạng của bộ đệm, đo thông lượng mạng,…) để lựa chọn mức tốc độ bit của phân

đoạn video tiếp theo. Khi các chunk đã được tải về thiết bị người dùng, sẽ được lưu trữ trong bộ đệm, được giải mã (decode) và sau đó hiển thị thông qua các chương trình chơi video (Ví dụ: VLC, KMPlayer như đã nói ở trên),lưu ý rằng phân đoạn muốn phát phải được tải xuống hoàn toàn.

Lịch sử của truyền trực tuyến có từ lâu và hình thức này được xem như lần đầu vào những năm 1890, đó là khi âm nhạc được phát trực tuyến thông qua mạng điện thoại . Tính đến 2020, thị trường phát trực tuyến có trị giá hàng tỉ đôla và ước tính tăng trưởng mở rộng hàng năm từ 21% từ năm 2021. Các nhà công nghệ khổng lồ, như là Facebook, Twitter và Youtube đầu từ mạnh mẽ và giành giật thị phần béo bở khổng lồ này.

- Phát trực tuyến video được sử dụng rộng rãi trong các ứng dụng mạng như: các phần mềm (các ứng dụng nghe nhạc, xem phim như VLC, KMPlayer; hay các trình duyệt web như: Internet Explorer, Google Chrome…) trên các máy khách truy cập và xem video từ các máy chủ theo mô hình máy chủ/máy khách; các ứng dụng họp trực tuyến, đào tạo từ xa.

Vì phát trực tuyến video đóng vai trò ngày một quan trọng trong mạng Internet nên đã có nhiều giao thức phát trực tuyến video được phát triển, phục vụ hiện nay, bao gồm:

- Real Time Transport Protocol (RTP)
- Real Time Messaging Protocol (RTMP)
- HTTP Live Streaming (HLS)
- Adobe HTTP Dynamic Streaming (HDS)
- IIS Smooth Streaming
- MPEG-DASH

Trong các giao thức trên, RTP và RTMP hoạt động tốt trong các mạng IP được quản lý. Tuy nhiên, trong Internet ngày nay, các mạng được quản lý đã được thay thế, nhiều mạng không hỗ trợ truyền phát RTP. Ngoài ra, các gói

RTP và RTMP thường không được phép thông qua tường lửa. Các giao thức còn lại đều dựa trên nền tảng HTTP.

Phát trực tuyến video là ứng dụng chiếm phần lớn lưu lượng Internet ngày nay. Các phương thức phát video ngày càng được cải thiện và nâng cao chất lượng. Bên cạnh đó, kết nối băng thông rộng cùng với sự phát triển của các thiết bị di động 3G/4G/5G, do đó, người dùng có thể sử dụng nhiều loại thiết bị khác nhau để truy cập kho nội dung đa phương tiện khổng lồ bằng nhiều phương thức kết nối với tốc độ truy cập Internet khác nhau. Tuy nhiên, cũng chính điều này đặt ra thách thức cho các nhà cung cấp dịch vụ trong việc đảm bảo người dùng nhận được các video với chất lượng cao và xem liên tục, không bị đứng hình.

Nhiều nghiên cứu đã chứng minh, người dùng sẽ ngừng xem các video khi có các khi có các vấn đề xảy ra, như lỗi ngay từ lúc khởi đầu xem video hoặc chuyển đổi từ mức chất lượng cao nhất sang chất lượng thấp nhất,… và ảnh hưởng nghiêm trọng đến thu nhập của các nhà cung cấp dịch vụ. Điều này bị ảnh hưởng từ nhiều yếu tố như chất lượng mạng, thiết bị đầu cuối và phương thức truyền.

Để giải quyết các vấn đề này, các nhà cung cấp dịch vụ nội dung triển khai và tối ưu các thuật toán tương thích tốc độ bit (Adaptive Bitrate algoritms – ABR algorithms) nhằm mục đích chính là nâng cao trải nghiệm người dùng (Quality of Experience – QoE) trong các điều kiện kết nối khác nhau để người dùng chủ động lựa chọn chất lượng các các phân đoạn video tiếp theo với mức QoE tốt nhất – dựa trên sự giám sát các điều kiện khả dụng như thông lượng mạng, tình trạng bộ đệm phát lại,…

1.1.2 *Vai trò của QoE và các yếu tố ảnh hưởng đến QoE*

Quality of Experience – QoE trải nghiệm người dùng là sự đánh giá cảm nhận của người dùng về chất lượng của dịch vụ, ở đây là chất lượng video mà người dùng nhận được khi sử dụng dịch vụ phát trực tuyến. Do có nhiều giao thức phát trực tuyến, nên việc đánh giá QoE khá khó khăn.

Lựa chọn tốc độ bit để tối ưu QoE là một nhiệm vụ đối mặt với nhiều khó khăn, thách thức khác vì có nhiều vấn đề mà một ABR phải đối mặt: (1) là sự biến đổi thông lượng mạng [12] (Zou et al., 2015), (2) mâu thuẫn giữa các tham số đo lường đánh giá QoE, như là các phân đoạn video phải có mức tốc độ bit cao hơn - đồng nghĩa các phân đoạn này có kích thước lớn hơn - đồng thời phải đảm hiện tượng rebuffer ở mức thấp nhất và (3) là sự phân cực trong khoảng thời gian dài, nghĩa là video được mã hóa và chia nhỏ thành nhiều phân đoạn, thuật toán ABR phải đảm bảo tối đa hóa tham số QoE cho tất cả các phân đoạn video này.

Có nhiều hàm định nghĩa các tham số đo lường QoE, nhưng có hai nguyên tố quan trọng nhất ảnh hưởng đến người dùng đã được nhiều tài liệu chứng minh, các yếu tố này quan trọng, ảnh hưởng trực tiếp đến người dùng, hầu như quyết định đến việc khách hàng có tiếp tục sử dụng dịch vụ hay không đó là chất lượng của các *chunk- phân đoạn video* mà khách hàng nhận được và tổng thời gian video bị đứng hình do hiện tượng *rebuffering.*

1.2. Kết luận chương

Chương này đã trình bày các cơ sở lý thuyết cần thiết khi nghiên cứu về phát trực tuyến video. Vai trò của QoE cũng như các yếu tố ảnh hưởng của nó đến quá trình phát trực tuyến. Chương tiếp theo sẽ trình bày các công trình nghiên cứu mà luận văn tham khảo, những công trình này đã góp phần định hướng nghiên cứu cho đề tài.

CHƯƠNG 2. CÁC THUẬT TOÁN LỰA CHỌN TỐC ĐỘ BIT TƯƠNG THÍCH TRONG VIDEO STREAMING

2.1. Tổng quan

Để đánh giá các hiệu quả của các giải pháp đã có, phải sử dụng một thước đo tổng thể để đánh giá chất lượng video vì trải nghiệm của người dùng bị ảnh hưởng mạnh mẽ bởi mức chất lượng nhận được. Với mức chất lượng cao hơn rõ ràng mang lại trải nghiệm tốt hơn và người dùng không hài lòng với chất lượng video suy giảm một cách đáng kể. Ngoài ra, giá trị QoE cũng có thể bị suy giảm đến mức tồi tệ do hiện tượng video bị gián đoạn thường xuyên vì sự biến động băng thông khả dụng.

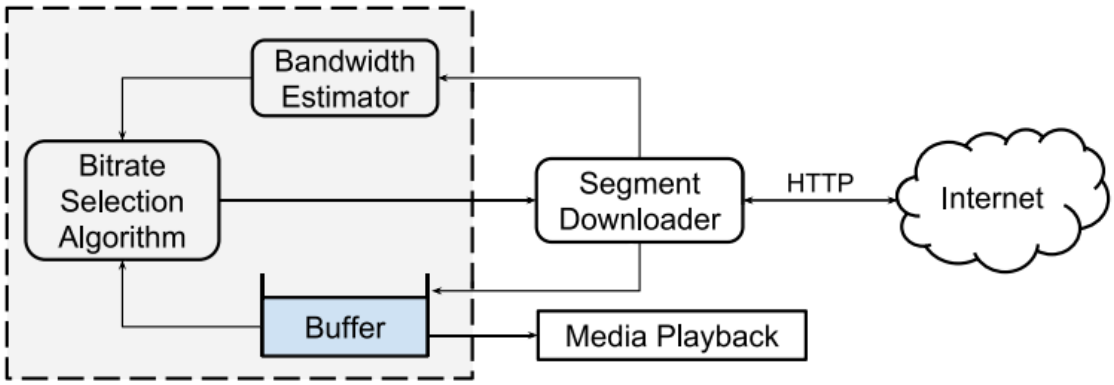
2.1.1 Các thuật toán tương thích tốc độ bit hiện có và xu hướng sắp tới

Trên thực tế, mục tiêu chính của các thuật toán tương thích tốc độ bit là nhằm tối ưu hóa chất lượng video nhận được tại người dùng, tối đa hóa QoE. Các thuật toán này được triển khai tại đầu cuối người dùng và tự động lựa chọn mức chất lượng của các phân đoạn video được tải tiếp theo dựa trên việc quan sát các thông số như ước lượng thông lượng mạng và tình trạng khả dụng của bộ đệm. Tuy nhiên, việc ước lượng này gặp nhiều thách thức do thông lượng biến động, mâu thuẫn trong các thông số đánh giá QoE (chất lượng cao, ít đứng hình và video phải mượt mà,…).

Các thuật toán tương thích tốc độ bit ban đầu có thể được phân thành hai lớp chính được mô tả như trong Hình 2.1: thuật toán dựa trên thông lượng mạng và thuật dựa trên bộ đệm. Và sau đó được phát triển thêm thành thuật toán kết hợp cả hai thuật toán cơ bản ban đầu.

Đối với nhóm thuật toán dựa trên thông lượng mạng, đầu tiên thuật toán sẽ ước tính thông lượng mạng khả dụng bằng cách sử dụng các thông số có thể thu thập được như chất lượng của phân đoạn đã tải xuống trước đó, lưu lượng mạng trước đó và sau đó yêu cầu mức chất lượng video cao nhất mà mạng được dự

đoán có thể xử lý. Ví dụ: dự đoán thông lượng dựa trên giá trị trung bình của thông lượng trước đó của một số phân đoạn đã được tải xuống. Mặc dù có nhiều nỗ lực nhằm cải thiện hiệu suất nhưng thực tế, thuật toán dựa trên thông lượng vẫn khó thực hiện



Hình 2.1. Sơ đồ mô phỏng các thuật toán ABR phổ biến ban đầu

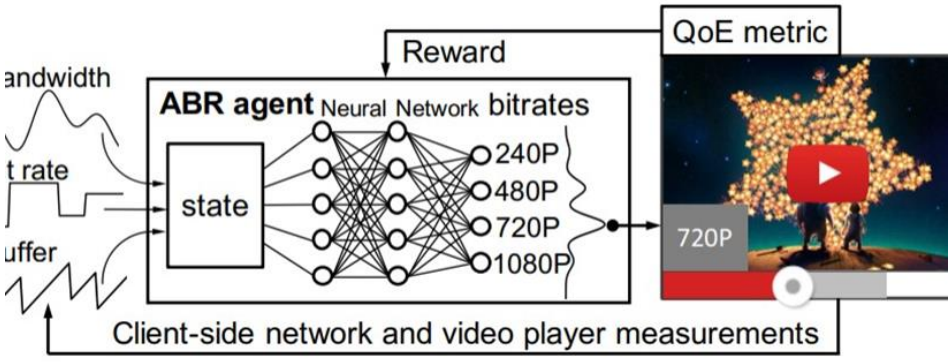
Các phương pháp dựa trên bộ đệm xem xét việc sử dụng bộ đệm phát lại của khách hàng khi quyết định mức chất lượng của của phân đoạn tiếp theo. Mục tiêu là giữ cho vùng đệm dưới một ngưỡng sao cho cân bằng giữa mức chất lượng và thời gian phát lại. Thuật toán BOLA tối ưu hóa giá trị QoE bằng cách sử dụng công thức tối ưu hóa Lyapunov. BOLA cũng hỗ trợ việc bỏ qua tải xuống phân đoạn tiếp theo, khi đó, trình phát video có thể tải lại một phân đoạn ở mức bitrate thấp hơn nếu nghi ngờ rằng sắp xảy hiện tượng đứng hình (rebuffer).

Bên cạnh các phương pháp độc lập, một số nghiên cứu nhằm mục đích kết hợp hai cách tiếp cận này: sử dụng kết hợp hai thông số thông lượng mạng và tình trạng bộ đệm để quyết định lựa chọn mức chất lượng của phân đoạn video tiếp theo. MPC là thuật toán điển hình cho nhóm này. Thuật toán MPC sử dụng các thuật toán điều khiển mô hình dự đoán sử dụng cả ước tính thông lượng và thông tin dung lượng bộ đệm để chọn chất lượng của phân đoạn tiếp theo được tải xuống, với mục tiêu chính vẫn là nhằm mang lại chất lượng video cao nhất cho người dùng, tối đa hóa giá trị QoE. Tuy nhiên, việc tính toán vẫn trên kết quả dự đoán, do đó, thuật toán MPC tồn tại nhược điểm lớn, đó là phụ

thuộc rất nhiều vào độ chính xác của kết quả dự đoán thông lượng, dẫn đến hiệu suất của thuật toán có thể bị suy giảm đáng kể nếu kết quả dự đoán không chính xác.

Một hướng nghiên cứu khác là áp dụng phương pháp học tăng cường (Reinforcement Learning: RL) để phát trực tuyến video. Các công trình của [14], [15], [16] sử dụng học tăng cường ở dạng bảng tìm kiếm, thay vì mạng nơ-ron. Đối với dạng bảng tìm kiếm, học tăng cường sẽ học hàm giá trị cho tất cả các kết hợp có thể có của các trạng thái và hành động rõ ràng, tuy nhiên, giải pháp này không thể áp dụng khi không gian trạng thái tăng lên. Pensieve là giải pháp áp dụng Deep RL, giải pháp này sử dụng mạng nơ-ron thay vì sử dụng các bảng tìm kiếm. Thuật toán lựa chọn tốc độ bit tương thích của Pensieve được tạo ra bằng cách sử dụng các quan sát về kết quả hiệu suất của các quyết định trước đây qua một số lượng lớn các thử nghiệm phát trực tuyến video. Điều này cho phép Pensieve tối ưu hóa chính sách của mình tùy thuộc vào các đặc điểm mạng khác nhau và tối ưu các tham số QoE một cách trực tiếp từ kinh nghiệm đạt được.

Từ những phân tích trên, chúng ta có thể thấy, các giải pháp truyền thống gần như dựa trên sự "dự đoán", và tùy thuộc vào kết quả của dự đoán sẽ thu được kết quả. Nếu kết quả dự đoán không chính xác, sẽ làm hỏng cả quá trình tính toán. Và từ những tồn tại đó, học tăng cường với những ưu điểm vượt trội đã được chứng minh trong các nghiên cứu gần đây trở thành xu hướng nghiên cứu chính trong việc tối ưu và nâng cao trải nghiệm người dùng trong dịch vụ phát trực tuyến video – dịch vụ đang chiếm phần lớn lưu lượng mạng Internet.



Hình 2.2: Áp dụng học tăng cường trong việc lựa chọn chất lượng video theo giải pháp Pensive

Hình 2.2 tóm tắt cách học tăng cường có thể được sử dụng để triển khai việc tương thích tốc độ bit trong phát trực tuyến video. Theo đó, chính sách hướng dẫn để thuật toán tương thích tốc độ bit đưa ra quyết định lựa chọn tốc độ bit của phân đoạn video tiếp theo được tải xuống không phải thực hiện một cách thủ công. Thay vào đó, quyết định của thuật toán có được từ việc huấn luyện một mạng nơ-ron. Tác nhân học tăng cường sẽ quan sát một tập hợp các chỉ số bao gồm trạng thái khả dụng của bộ đệm tại phía người dùng, các quyết định về tốc độ bit trước đó và một số thông tin về tình trạng mạng (ví dụ: các phép đo thông lượng) và cung cấp các giá trị này cho mạng nơ-ron làm dữ liệu đầu vào, dữ liệu đầu ra thu được là quyết định lựa chọn tốc độ bit của phân đoạn video tiếp theo được tải xuống. Kết quả QoE sau đó được quan sát và chuyển trở lại cho tác nhân ABR như một phần thưởng. Tác nhân sử dụng chính thông tin phần thưởng này để huấn luyện và cải thiện mô hình mạng nơ-ron của nó.

2.2. QoE và cách đánh giá QoE

Như đã nói ở trên, Quality of Experience – QoE - trải nghiệm người dùng- là sự đánh giá cảm nhận của người dùng về chất lượng của dịch vụ, ở đây là chất lượng video mà người dùng nhận được khi sử dụng dịch vụ video trực tuyến. Theo yêu cầu thực tế, QoE càng cao càng tốt. Tuy nhiên, QoE bị ảnh hưởng bởi nhiều yếu tố khác nhau nên việc xây dựng công thức cho QoE cũng là một thách thức lớn

2.2.1 *Công thức QoE phát trực tuyến video*

Đối với video streaming, một video được chia thành nhiều phân đoạn N với thời lượng bằng nhau τ. Mỗi phân đoạn được được mã hóa với các mức chất lượng L khác nhau và được phân bố thành các luồng riêng lẻ với các cấp độ và tên quen thuộc: như 720p, 1080p, 1080p @ 30fps. Đối với các phân đoạn có cùng chỉ số n, mức chất lượng cao hơn đồng nghĩa với kích thước lớn hơn. ($\sigma(n, l_1) < \sigma(n, l_2)$ và ($q(n, l_1) < q(n, l_2)$, $l_1 < l_2$. Theo đó, người dùng

yêu cầu một phân đoạn từ máy chủ và phân đoạn được tải xuống máy khách. Các đoạn đã tải xuống được phát lại tại phía người dùng.

Thay vì ghi trực tiếp phân đoạn đã tải vào bộ nhớ của máy khách, phân đoạn tải xuống sẽ được lưu trữ trong "Replay Buffer" – bộ đệm phát lại Ω trước khi được phát, sau đó sẽ được lưu trữ trong RAM (Bộ nhớ truy cập ngẫu nhiên). Kích thước bộ đệm phát lại được xác định trước tùy thuộc vào máy khách, nhưng thông thường nó có thể kéo dài hàng chục giây. Bộ đệm phát lại sử dụng quy tắc first-in-first-out: các phân đoạn được tải xuống trước sẽ được phát trước. Phân đoạn phải được tải xuống đầy đủ vào bộ đệm phát lại và mới bắt đầu phát.

Trình phát video yêu cầu mức đệm phát lại ban đầu Ω_{min} trước khi bắt đầu phát, tức là các phân đoạn P được tải xuống từ trước trước và sẽ không được sử dụng trong mục tiêu tối ưu hóa (thường là $P = 1$). Khi bộ đệm phát lại vượt quá mức ngưỡng trên Ω_{max}, trình phát video sẽ dừng yêu cầu các phân đoạn mới. Người dùng phải đợi mức bộ đệm giảm xuống dưới Ω_{max} để gửi lại yêu cầu phân đoạn mới. Trình phát video bị hiện tượng rebuffers (đứng hình) khi chỉ số phân đoạn sắp được phát tiếp theo không có sẵn trong bộ đệm. Sau đó, trình phát video tạm dừng phát và đợi phân đoạn mới cho đến khi nó được tải xuống hoàn toàn và gây ra hiện tượng đứng hình. Theo đánh giá, hiện tượng đứng hình khi đang xem tác động mạnh mẽ đến trải nghiệm chất lượng của người dùng.

Các thuật toán tương thích tốc độ bit nhằm tối ưu hóa chất lượng trải nghiệm QoE của người dùng trong các điều kiện khác nhau để tự động chọn chất lượng cho từng phân đoạn (là các block 4 giây như đã nói ở trên) dựa trên việc quan sát độ khả dụng, chẳng hạn như ước tính thông lượng mạng và kích thước của bộ đệm phát lại, nhằm giảm hiện tượng đứng hình.

Có nhiều phương pháp [6], [12] đánh giá chất lượng của các thuật toán tương thích tốc độ bit trong việc cải thiện và nâng cao giá trị QoE, điểm chung là các phương pháp này tập trung vào hai yếu tố chính đó là chất lượng tổng của đoạn video mà người dùng nhận được và thời lượng video bị đứng hình.

Nghĩa là tối đa tổng các giá trị cực đại chất lượng q của video được tải xuống, trong khi vẫn đảm bảo video được phát liên tục, không bị gián đoạn (tức là phân đoạn video thứ *n* phải được tải xuống hoàn toàn trước khi phân đoạn video thứ *n-1* phát xong) nhưng không giới hạn kích thước bộ đệm phát lại (tức là $\Omega_{max} = \infty$).

Theo [6], [12], công thức cho hàm QoE được tính như sau:

$$\sum_{n=1}^{N} q(R_n) - \alpha \sum_{n=1}^{N} T_n - \beta \sum_{n=1}^{N} |q(R_n) - q(R_{n-1})| \quad (1)$$

Trong biểu thức (1) gồm có:

- N là tổng số phân đoạn (chunk) của video
- R_n là tốc độ bit của phân đoạn thứ n
- $\boldsymbol{q(R_n)}$ là hàm độ lợi tương ứng với giá trị tốc độ bit R_n của phân đoạn thứ *n* với mức chất lượng người dùng nhận được. Giá trị $\boldsymbol{q(R_n) = R_n}$
- T_n là thời gian đứng hình
- $\boldsymbol{|q(R_n) - q(R_{n-1})|}$ là độ sai lệch mức chất lượng của hai phân đoạn liền kề.
- α *v*à β là các hệ số giảm trừ do lỗi đứng hình và lỗi chuyển đổi mức chất lượng tương ứng. Giá trị $\alpha = 2.66$ và $\beta = 1$ được sử dụng theo [6]

Từ công thức (1) ta có thể thấy, nhằm nâng cao QoE, nâng cao chất lượng trải nghiệm người dùng, đó là nâng cao tổng chất lượng video nhận được, giảm thiểu hiện tượng đứng hình và giảm chuyển đổi mức chất lượng video tại các thời điểm tương ứng, và đây cũng chính là mục tiêu mà các thuật toán ABR hướng đến.

2.4. Kết luận chương

Trong chương này thông qua việc nghiên cứu tìm hiểu được một số thuật toán hiện có cũng như xu hướng trong tương lai, đồng thời cũng trình bày công thức về QoE cho phát video trực tuyến. Tạo nên tiền đề và cơ sở vững chắc cho nghiên cứu của đề tài luận văn này.

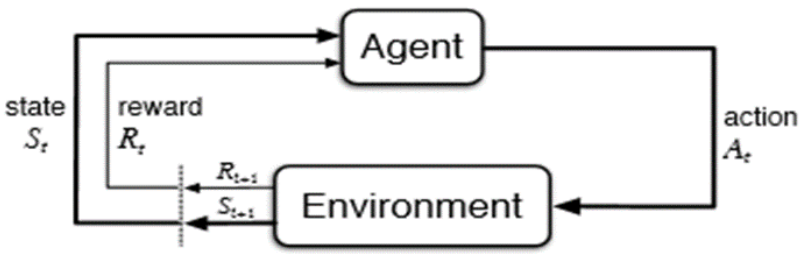
CHƯƠNG 3. GIẢI PHÁP NÂNG CAO CHẤT LƯỢNG PHÁT TRỰC TUYẾN VIDEO: HỌC TĂNG CƯỜNG (REINFORCEMENT LEARNING)

3.1. Phương pháp học tăng cường

3.1.1. Tổng quan về học tăng cường

Có rất nhiều giải pháp cho phát trực tuyến video, với mục tiêu chính là nâng cao chất lượng QoE, đem lại cho người dùng trải nghiệm tốt nhất. Tuy nhiên, như đã nói, chất lượng thu được của các giải pháp hiện có tùy thuộc vào kết quả dự đoán. Nếu kết quả dự đoán sai, kết quả thu được có thể không tốt, dẫn đến chất lượng video kém, kéo giảm giá trị QoE. Và từ đó, để khắc phục các hạn chế của các giải pháp trước đó, giải pháp học tăng cường được đề xuất và đã chứng minh được hiệu quả triển khai thực tế.

Học tăng cường là việc huấn luyện các mô hình học máy để đưa ra một chuỗi các quyết định. Trong học tăng cường, sử dụng một tác nhân (agent) tương tác với môi trường (environment). Tại thời điểm *t*, tác nhân lấy thông tin từ môi trường để tìm ra trạng thái s_t , từ đó tác nhân sẽ thực hiện hành động a_t. Tác nhân sẽ nhận được phần thưởng (reward) r_t tương ứng với hành động a_t, trong khi trạng thái của môi trường thay đổi từ s_t sang s_{t+1}. Giá trị của r_t sẽ cho biết tình trạng hiện tại của môi trường là tốt hay xấu. Mục tiêu chính của nó là tối đa phần thưởng tổng, gọi là lợi tức. Học tăng cường là cách để tác nhân học các thao tác này và đạt được mục tiêu đề ra.



Hình 3.1: Sơ đồ tổng quan RL

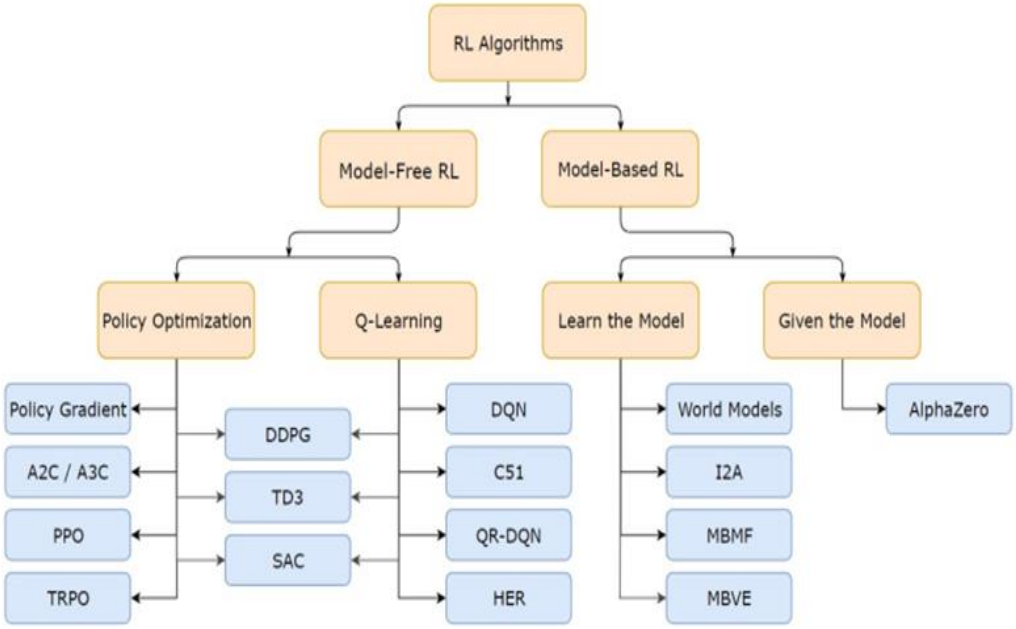
Hình 3.1 mô tả cách tác nhân thu thập trạng thái s_t của môi trường, thực hiện hành động a_t và thu được phần thưởng r_t.

Hai thành phần chính của học tăng cường là tác nhân và môi trường. Môi trường là nơi tác nhân tồn tại và tương tác. Ở mỗi bước tương tác, tác nhân sẽ quan sát và thu thập thông tin tình trạng của môi trường và quyết định hành động tiếp theo. Môi trường có thể thay đổi khi có tác nhân tác động hoặc tự thay đổi, không cần tác động nào.

Để hiểu rõ hơn thể hơn, chúng ta cần giới thiệu và làm rõ một số thuật ngữ sử dụng trong học tăng cường:

- không gian trạng thái,
- không gian hành động,
- chính sách,
- quỹ đạo,
- phần thưởng và lợi tức,
- và các hàm giá trị: Q-funtion, V-function.

3.1.2. The RL Landscape: Các mô hình RL



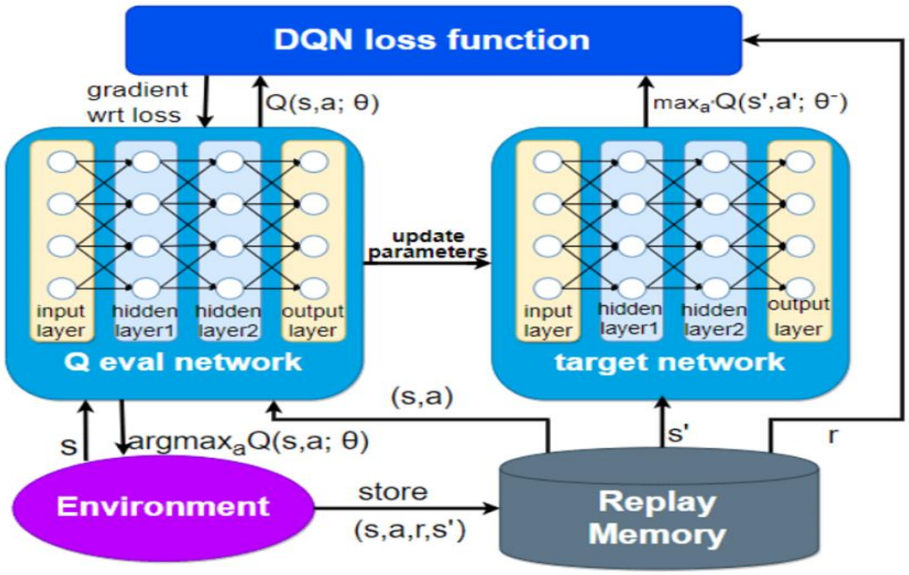
Hình 3.2: Các mô hình RL

3.2. Q-Learning và Deep Q-Learning

3.2.1. *Q-Learning*

3.2.2. *Deep Q-Learning*

viết tắt là DQN. DQN là thuật toán hiện đại trong họ Q-Learning, là sự kết hợp của phương pháp học sâu và Q-Learning, và khi triển khai trên DASH nhằm đạt được chính sách tối ưu cho mô-đun điều giao thức tương thích DASH. Hệ thống học máy này đã được sử dụng trong các hệ thống phức tạp trong các công trình nghiên cứu và thể hiện hiệu suất vượt trội, dù phương phát này mới xuất hiện gần đây.



Hình 3.3: Sơ đồ hoạt động của DQN

Hình 3.3 mô tả sơ đồ hoạt động của giải pháp học tăng cường DQN. Nếu xem nội dung video dưới dạng một chuỗi các cảnh với thời lượng được phân phối theo cấp số nhân, dịch vụ phát trực tuyến video có thể được mô hình hóa như một một chuỗi quyết định Markov với không gian hành động *A*, không gian trạng thái *S* và hàm phần thưởng ρ*: S* ×*S* × *A* → *R*. Tương ứng, ở đây sử dụng *qt* để biểu thị hành động tải xuống một phân đoạn *t* với chất lượng hình ảnh *qt*. Hành động *qt* ∈*A*, được lấy khi hệ thống ở trạng thái cho trước *st* ∈*S*, xác định phân phối thống kê của trạng thái tiếp theo *st + 1* và phần thưởng ρ *(st, st + 1, qt)* đạt được ở bước *t*. Hàm tổn thất $\tilde{L}$ tại bước t được đánh giá thông qua bộ bốn tham số $e_t = (s_t, q_t, r_{t,} s_{t+1})$, được xem như là trải nghiệm của tác nhân tại bước t và được tính như sau:

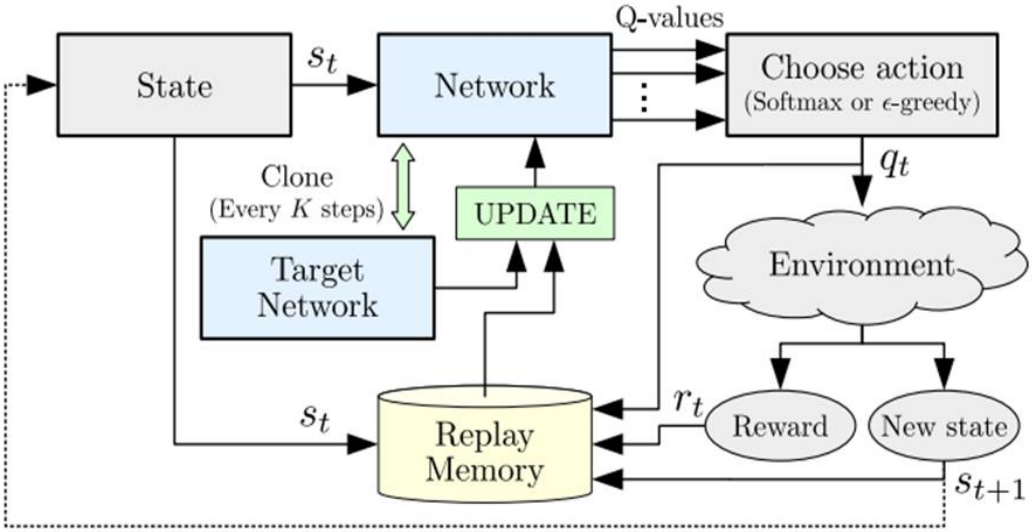
$$\tilde{L}(s_t, q_t, r_{t,} s_{t+1}) = \left(r_t + \alpha \max_q \hat{Q}\left(s_{t+1}, q_t | \overline{w_t}\right) - Q(\langle s_t, q_t | w_t \rangle\right)^2 \quad (8)$$

Với r_t là phần thưởng tại phân đoạn *t*. Đối với DASH, khi triển khai DQN, có hai mạng nơ-ron sâu được sử dụng. Mạng thứ nhất, với vector trọng số w_t, được cập nhật sau mỗi phân đoạn (thường là sau bước thời gian *t*), và được dùng để xây dựng bảng giá trị Q-value $Q(\langle s_t, q_t | w_t \rangle$. Mạng thứ hai, thường gọi là mạng đích, được sử dụng nhằm tăng tính ổn định của hệ thống học máy và vector trọng số $\overline{w_t}$ được cập nhật sau K phân đoạn, được thiết lập bằng với giá trị của mạng thứ nhất và được giữ cố định cho K-1 bước tiếp theo. Nghĩa là $w_t = \overline{w_t}$ sau K phân đoạn. Mạng đích được dùng để tìm kiếm giá trị $Q(\langle s_t, q_t | \overline{w_t} \rangle$.

Ta định nghĩa mạng nơ-ron sử dụng dữ liệu đầu vào là các trạng thái và dữ liệu đâu ra là các giá trị *Q-value*. Thế nhưng mạng nơ-ron dễ bị hiện tượng tràn nếu liên tục nhận các trạng thái giống nhau hoặc có tính tuyến tính, khi đó cần phải áp dụng kỹ thuật Experience Replay để tăng tính ổn định của thuật toán và tận dụng các dữ liệu đã thu thập trước đó.

Thay vì với mỗi trạng thái đầu vào, mạng nơ-ron sẽ cập nhật một lần, ta lưu lại các trạng thái này vào bộ nhớ replay-memory. Sau đó thực hiện lấy mẫu các trạng thái này thành các batch đưa vào mạng nơ-ron và thực hiện việc huấn luyện. Việc này giúp đa dạng hóa dữ liệu đầu vào và tránh mạng nơ-ron bị quá tải. Tuy nhiên, bộ nhớ để lưu trữ và các mẫu này cũng cần phải đủ lớn để giảm sự biến động.

Điều này đem lại các lợi ích sau đây: dữ liệu đáng tin cậy hơn, các mẫu huấn luyện ít bị trùng lắp, chính sách và quá trình lấy mẫu độc lập, không phụ thuộc.



Hình 3.4: Lưu đồ tiến trình cập nhật

Toàn bộ quá trình có thể được chia thành 2 giai đoạn liên tiếp như Hình 3.4, thực thi khác nhau nhưng có cùng số bước thực hiện, được gọi là giai đoạn huấn luyện và giai đoạn kiểm thử.

Giai đoạn huấn luyện: Tham số thăm dò, cụ thể là ε trong trường hợp của chính sách *epsilon* ε-tham lam được giảm dần. Ở mỗi lần lặp, trọng số mạng được cập nhật để giảm thiểu hàm tổn thất trong (8). Phương pháp Adam được sử dụng làm thuật toán tối ưu hóa gradient giảm dần: thực thi tốc độ học tập tương thích để việc hội tụ diễn ra nhanh hơn.

Giai đoạn kiểm thử: Tham số thăm dò được đặt thành 0, do đó, chính sách *epsilon* ε-tham lam thực hiện các hành động được coi là tối ưu tương ứng với trạng thái hệ thống hiện tại và ánh xạ *Q (st, qt | wt)* từ mạng nơron đầu tiên. Đối với giai đoạn này, trọng số *wt* đã bị đóng băng và không còn được cập nhật trong suốt thời gian kiểm tra. Mạng mục tiêu không được sử dụng trong giai đoạn kiểm tra và tất cả các đánh giá hiệu suất đều dựa trên kết quả thu được trong giai đoạn thứ hai này.

Sơ đồ quá trình cập nhật được hiển thị trong Hình 3.3. Đầu tiên, trạng thái hiện tại của môi trường s_t được đưa vào mạng nơ-ron, kết quả đầu ra là giá trị dự đoán Q cho mỗi hành động có thể có $q \in A$, tức là, các giá trị khác nhau của tập tương thích *A*. Sau đó, một hành động q_t được chọn theo chính sách ε-

tham lam hoặc softmax. Khi thực hiện hành động a_t, hệ thống chuyển sang trạng thái mới s_{t+1} và phần thưởng mới r_t được đánh giá theo công thức:

$$r_i = q(l_i) - \beta|q(l_i) - q(l_{i-1})| - \gamma\emptyset_i - \delta[\max(0, B^{max} - Bi\,]]^2 \quad (9)$$

Trong đó:

- $q(l_i)$ là hàm độ lợi, tương ứng mức chất lượng l_i của phân đoạn video thứ i.
- $|q(l_i) - q(l_{i-1})|$ là độ sai lệch mức chất lượng của hai phân đoạn video liên tiếp. Mức chất lượng của video được xem là ổn định khi độ sai lệch bằng 0 hoặc rất nhỏ. Các phân đoạn video nhận được có sự thay đổi liên tục mức chất lượng sẽ ảnh hưởng nghiêm trọng đến cảm nhận của người dùng.
- $\emptyset_i$ là thời gian bị đứng hình khi phát phân đoạn thứ i, $\emptyset_i$ được tính theo công thức $\emptyset_i = \max(0, d_i - B_i)$, với d_i là thời gian tải phân đoạn thứ i và B_i là kích thước của bộ đệm (tính theo giây).
- $[max\,(0, B^{max} - Bi\,]^2$ giảm trừ khi bộ đệm video có giá trị thấp hơn mức ngưỡng cho trước B^{max} của bộ đệm. Tuy nhiên, giá trị này có thể bỏ qua trong công thức QoE.
- β, γ và δ là các hệ số cho các thành phần giảm trừ do đứng hình, giảm trừ do thay đổi mức chất lượng và giảm trừ khi mức bộ đệm thấp hơn ngưỡng cho trước.

Thuật toán DQN được mô tả như sau:

Initialize replay memory R with fixed capacity

Initialize action-value function $\hat{a}$ with random weights w

Initialize target action-value function $\hat{q}$ with weight $w_{tar} = w$

For episode $m = 1, \ldots, M$ **do**

For time step $t = 1, \ldots, N$ **do**

Select action $a_t = \begin{cases} \text{random action, with probability } \epsilon \\ \arg\max_{a'} \hat{q}(s_t, a'; w), \quad \text{otherwise} \end{cases}$

Take action a_t and observe reward r_t and new state s_{t+1}

Append transition (s_t, a_t, r_t, s_{t+1}) to R

Sample uniformly a random mini-batch of B transitions $(\boldsymbol{s}_j, \boldsymbol{a}_j, \boldsymbol{r}_j, \boldsymbol{s}_{j+1})$ from R_k

Set $y_j = \begin{cases} r_j & \text{for terminal step } j+1 \\ r_j + \gamma \max\limits_{a'} \hat{q}(s_{j+1}, a'; w_{tar}) & \text{for non} - \text{terminal step } j+1 \end{cases}$

Perform a stochastic gradient descent step w.r.t. loss function

$$J(w) = \frac{1}{B}\sum_{j=1}^{B}\left(y_j - \hat{q}(s_j, a_j; w)\right)^2$$

Every fixed C steps, update target network $w_{tar} = w$

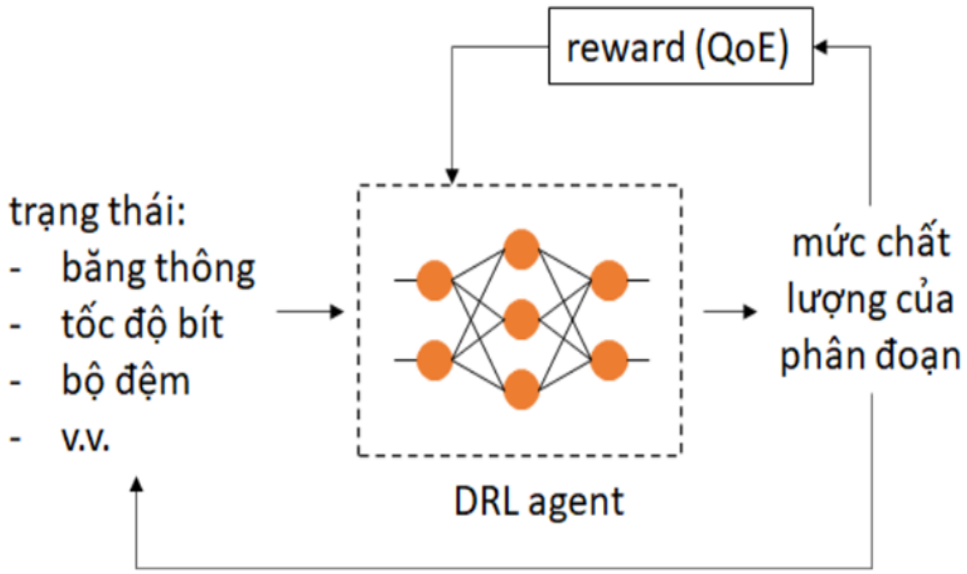
End for

End for

3.3. Ứng dụng DQN đối với Video Streaming

Như đã nói ở trên, DQN là phương pháp học kết hợp học tăng cường Q-Learning với giải pháp học sâu, sử dụng các mạng nơ-ron, và thực hiện học mô hình thông qua một tập hợp các hành động do tác nhân học tăng cường đưa ra.

Tại thời điểm *t*, tương ứng với trạng thái môi trường *st,* khi tác nhân thực hiện hành động a_t, sẽ tương tác với môi trường, môi trường sẽ chuyển sang trạng thái s_{t+1} , và nhận được phần thưởng r_{t+1} . Mục tiêu của việc học là đưa ra chuỗi hành động nhằm đạt được giá trị tối đa của tổng phần thưởng nhận được.



Hình 3.5: Mô hình học tăng cường cho vấn đề phát video tương thích tốc độ bit qua HTTP

Khi áp dụng giải pháp DQN vào phát trực tuyến, như Hình 3.5, không gian trạng thái , các các hàm phần thưởng, hành động, hàm phần thưởng, tác nhân học tăng cườn được định nghĩa như sau:

Trạng thái (tương ứng với giá trị st) được định nghĩa là tập hợp các quan sát từ môi trường hiện tại như ước tính thông lượng mạng, độ trễ, chất lượng của các phân đoạn video vừa tải về trước đó, kích thước của phân đoạn tiếp theo tương ứng với các mức chất lượng khác nhau, số phân đoạn video còn lại,…

Hành động (tương ứng với giá trị at): hành động được định nghĩa là lựa chọn chất lượng phân đoạn video tiếp theo, tùy thuộc vào kết quả của việc quan sát trạng thái của môi trường.

Tác nhân học tăng cường (DRL agents) trong hướng nghiên cứu của luận văn là thuật toán DQN.

Hàm phần thưởng (reward) là giá trị QoE tổng thu thập được, là sự tổng hợp giữa độ lợi mang lại từ chất lượng của các phân đoạn video liên tiếp, giá trị này sẽ bị giảm trừ nếu hai phân đoạn liên tiếp có mức chất lượng khác nhau và giảm trừ khi bị đứng hình. Theo đó, hàm phần thưởng của phân đoạn video thứ *i* được tính theo công thức (9)

Sau quá trình huấn luyện bằng cách sử dụng thuật toán DQN, kết quả thu về là giá trị QoE tính toán được từ các hành động lựa chọn mức chất lượng của phân đoạn video tải tiếp theo.

3.4 Kết luận chương 3

Chương 3 đã nêu lên vấn đề mà luận văn sẽ đối mặt và đề xuất quy trình nghiên cứu. Trong chương sau, luận văn sẽ trình bày quá trình cụ thể quá trình xây dựng và đánh giá kết quả đạt được.

CHƯƠNG 4. CÀI ĐẶT VÀ THỰC NGHIỆM

4.1. Công cụ mô phỏng

Từ công thức đánh giá QoE và kết quả chương 3, luận văn tập trung xây dựng công cụ mô phỏng bằng việc sử dụng mã nguồn mở như Pytorch, Stable_Baseline 3 và OpenAI Gym.

4.1.1. PyTorch

PyTorch [18] (Paszke et al., 2019) là một framework học máy mã nguồn mở , giúp tăng tốc lộ trình từ các mẫu nghiên cứu đến triển khai thực tế. PyTorch cung cấp hai tính năng cao cấp: (1) Tính toán tensor (giống như NumPy) nhưng với khả năng tăng tốc mạnh mẽ thông qua GPU và (2) Mạng Deep neural được xây dựng trên hệ thống phân biệt tự động theo phân loại. PyTorch đang thịnh hành trong cộng đồng nghiên cứu do tính năng động của nó và hầu hết thư viện RL được xây dựng trên PyTorch cho phép toàn quyền tùy chỉnh..

4.1.2. *OpenAI Gym Environment*

Gym [19] (Brockman và cộng sự, 2016) là một bộ công cụ để phát triển và so sánh các thuật toán Reinforcement Learning. Hỗ trợ dạy các tác nhân mọi thứ, từ đi bộ đến chơi trò chơi như Pong hoặc Pinball. Nó không có giả định nào về cấu trúc của tác nhân và có thể tương thích với bất kỳ thư viện số tính toán nào..

4.1.3. Stable_Baseline 3

Stable_Baseline3 [20](SB3) (Rafn và cộng sự, 2021) là một framework mã nguồn mở triển khai các thuật toán deep RL không theo mô hình đã được sử dụng phổ biến. Thư viện rất chú trọng đến tuân thủ các phương pháp hay nhất về kỹ thuật phần mềm để đạt được các triển khai chất lượng cao phù hợp với các kết quả trước đó. Mỗi thuật toán đã được chuẩn hóa trên các môi trường chung và so với các lần triển khai trước. Bộ thử nghiệm bao gồm 95% mã và cùng với với các thay đổi đang xem xét kỹ lưỡng dựa trên người dùng đang

hoạt động, đảm bảo rằng bất kỳ lỗi triển khai nào được giảm thiểu. Vào tháng 11 năm 2021, SB3 có hơn 800 sao trên GitHub, hơn 100 vấn đề đã được xử lý và hơn 80 yêu cầu hợp nhất, làm cho SB3 trở thành một trong những thư viện RL phổ biến nhất.

4.2. Tập dữ liệu dùng cho quá trình mô phỏng

Đối với video, quá trình mô phỏng sử dụng tập dữ liệu video *Elephants dream* [15] (Blender, 2014). Video được mã hóa thành 20 mức chất lượng khác nhau với mỗi phân đoạn có thời lượng 4 giây. Bảy mức mã hóa bitrate sau: [700, 900, 2000, 3000, 5000, 6000, 8000] Kbps được chọn, tuân theo các cấu trúc của (Google, 2021), là các mức chất lượng phổ biến, thân thuộc với người dùng là: (240p, 360p, 480p, 720p, 720p @ 60fps, 1080p, 1080p @ 60 khung hình / giây). Do đó, tác nhân có 7 hành động riêng biệt cho mỗi bước. 60 phần đầu tiên của video (N = 60) được sử dụng, có thời lượng 240 giây. Chất lượng mặc định của phân đoạn đầu tiên là mức chất lượng thấp nhất.

4G LTE: tập dữ liệu 4G LTE [16](Raca và cộng sự, 2018) bao gồm 135 đoạn băng thông, với mức trung bình có thời lượng 15 phút cho mỗi đoạn băng thông, ở mức độ chi tiết 1 giây. Tập dữ liệu này đã thu thập đoạn băng thông từ các nhà khai thác di động Ireland, với 5 kiểu di chuyển (tĩnh, người đi bộ, xe hơi, xe buýt và xe lửa).

FCC: Tập dữ liệu FCC chứa hơn 1 triệu đoạn, ở mức độ chi tiết 10 giây mỗi mẫu [17] (FCC, 2019). Tôi tạo 1.000 đoạn băng thông ngẫu nhiên (mỗi đoạn kéo dài 320 giây) cho tập dữ liệu huấn luyện và kiểm thử của chúng tôi. Tôi sử dụng tập dữ liệu trong tháng 9 năm 2019.

4.3. Quá trình mô phỏng

Huấn luyện và kiểm thử: Trong cả hai tập dữ liệu, tôi chia ngẫu nhiên tập dữ liệu thành 80% cho huấn luyện và 20% để kiểm tra. Để tăng tốc quá trình huấn luyện, tôi kết hợp bộ dữ liệu FCC và LTE để huấn luyện tác nhân học tăng cường. Tác nhân được huấn luyện trong 590.000 bước với 10000 tập

để tìm ra mô hình. Trong quá trình huấn luyện, các mô hình tốt hơn so với mô hình trước đó sẽ sẽ được giữ lại để so sánh và tìm ra mô hình tốt nhất.

```
#Training DQN và lưu mô hình tốt

class DQNEvaluator(BaseEvaluator):
    def evaluate(self, file_name, params, save=True):
        self.name = "DQN"
        self.env = Monitor(self.env, file_name)
        self.env.reset()
        self.model = DQN("MultiInputPolicy", self.env, verbose=1, **params, tensorboard_log=logdir,device=device)
        for i in range(10):
            self.model.learn(total_timesteps=NUM_STEP_PER_EP * self.EVAL_EPS, reset_num_timesteps=False,tb_log_name = logdir)
            train_reward=self.env.get_episode_rewards()[-self.EVAL_EPS]
            train_reward += train_reward
            print(train_reward)
            if best_reward < train_reward and i > (2*10/3):
                best_reward == train_reward
                self.model.save(f"{model_dir}/{self.EVAL_EPS * i}")
```

Hình 4.1: Đoạn code huấn luyện và lưu các mô hình tốt

```
class DQNEvaluator(BaseEvaluator):
    def evaluate(self, file_name, params, save=True):
        self.name = "DQN"
        self.env = Monitor(self.env, file_name)
        self.env.reset()
        model_dir = "models/DQN"
        model_path = f"{model_dir}/1800.zip"
        self.model = DQN.load(model_path, self.env,verbose=1, **params, tensorboard_log=logdir,device=device, tb_log_name= lo

    def test(self, file_name, bitrate_list_test_fcc, bitrate_list_test_lte):
        print("Evaluating on FCC...")
        env = SinglepathEnvGym(bitrate_list=bitrate_list_test_fcc, train=False)
        info_keywords = ("reward_quality_norm", "reward_smooth_norm", "reward_rebuffering_norm")
        env = Monitor(env, filename="./test_monitorFCC" + self.name, info_keywords=info_keywords)
        avg_reward, _ = evaluate_policy(self.model, env, n_eval_episodes=config.system_config["EVAL_EPS"],
                                        return_episode_rewards=True)
        print(f"Algorithm evaluation average reward: {avg_reward}")  # Kieu cu
        df = pd.DataFrame({"eps_id": np.arange(len(avg_reward)), "avg_reward": avg_reward})
        # df.to_csv("./test_monitorFCC" + self.name + ".monitor.csv")
        df.to_csv("./test_monitorFCC" + self.name + ".csv")
```

Hình 4.2: Code đánh giá kết quả thu được theo tập dữ liệu test FCC

Sau khi huấn luyện, tác nhân DQN được đánh giá kiểm tra trên một tập thử nghiệm (được phân chia như mô tả ở trên) trong 200 tập và ghi lại giá trị phần thưởng trung bình. Thử nghiệm được lặp lại 10 lần và sử dụng giá trị trung bình. Dữ liệu đầu vào được cố định trong quá trình đánh giá, tức là tại mỗi bước, thuật toán quan sát dữ liệu đầu vào là như nhau.

Các thư viện mã nguồn mở: Sử dụng các thuật toán đã được triển khai trong thư viện mã nguồn mở Stable-Baselines3 mà không sửa đổi bất kỳ phần nào của thuật toán.

α và β: theo [6], đặt $\beta = 1$ và sử dụng giá trị $\alpha = 2.66$ để kiểm tra.

4.4. Đánh giá kết quả mô phỏng

4.4.1. Các thuật toán khác

So sánh phương pháp tương thích tốc độ bit dựa trên học tăng cường, ở đây là DQN, so với các thuật toán đã có trước đó nhận được kết quả như sau:

- Ngẫu nhiên (RAN): với thuật toán này, tại mỗi bước, mức chất lượng video được lựa chọn một cách ngẫu nhiên.
- Cố định (CON): thuật toán này sẽ chọn mức chất lượng như nhau tại mỗi bước, cụ thể là 3000kpbs, tương đương chuẩn video HD 720p.
- Dựa trên thông lượng (TRB): Mức chất lượng cao nhất được chọn nhưng phải nhỏ hơn bình quân mức chất lượng của ba phân đoạn được tải xuống gần nhất.
- BOLA: thuật toán tương thích dựa trên thông lượng, sử dụng phương pháp tối ưu Lyapunov để giảm thiểu đứng hình và tối ưu hóa chất lượng video.

4.4.2. Kết quả mô phỏng

Kết quả được thể hiện như trong Bảng 4.1 khi giá trị $\alpha = 2.66$ và bộ siêu tham số được lựa chọn sẵn. Thuật toán DQN có thể hội tụ sau 250.000 bước huấn luyện.

Hình 4.3: Biểu đồ giá trị phần thưởng tích lũy của DQN khi huấn luyện

Khi so sánh QoE của giải pháp QoE với các thuật toán của các giải pháp khác khác, thuật toán dựa trên DQN đem lại giá trị QoE cao nhất.

Bảng 4.1: Kết quả QoE khi thực hiện đánh giá với $\alpha = 2.66$

FCC	QoE	Chuyển đổi mức chất lượng	Rebuffer (Đứng hình)
DQN	**0.821**	0.19	0.06
THRB	0.726	0.20	0.03
BOLA	0.785	0.11	0.09
RAN	-1.142	0.606	1.38
CON	-1.686	0.044	2.8

LTE	QoE	Chuyển đổi mức chất lượng	Rebuffer (Đứng hình)
DQN	**0.485**	0.17	0.141
THRB	0.417	0.186	0.208
BOLA	0.455	0.152	0.265
RAN	-2.2005	0.604	2.380
CON	-3.14	0.044	4.251

CHƯƠNG 5. KẾT LUẬN

5.1. Kết quả nghiên cứu của đề tài

Luận văn "NÂNG CAO CHẤT LƯỢNG PHÁT VIDEO QUA HTTP BẰNG PHƯƠNG PHÁP HỌC TĂNG CƯỜNG" đã giới thiệu về lịch sử của phát video trực tuyến và các giải pháp hiện có. Tiếp theo tôi phân tích các yếu tố tác động đến chất lượng dịch vụ, tác động đến trải nghiệm người dùng và đánh giá các tác động này. Sau cùng, tôi đề xuất giải pháp, là các thư viện và các framework được dùng để mô phỏng, đánh giá kết quả thu được. Kết quả mô phỏng đã chứng minh tính hiệu quả của giải pháp học tăng cường sâu DQN khi áp dụng cho thuật toán tương thích tốc độ bit. Với kết quả là thuật toán tương thích tốc độ bit dựa trên học tăng cường thể hiện ưu điểm so với các phương pháp truyền thống.

5.2. Hạn chế của luận văn

Môi trường thực: Do quỹ thời gian hạn hẹp, tôi chỉ thực hiện việc đánh giá thông qua kết quả mô phỏng và sử dụng một thuật toán áp dụng học tăng cường để so sánh với các thuật toán truyền thống mà mà không thực hiện việc mô phỏng trong môi trường thực như dash.js. Trong môi trường thực sẽ có nhiều vấn đề hơn cần để giải quyết.

5.3. Vấn đề kiến nghị và hướng đi tiếp theo của nghiên cứu

Từ kết quả thực tế và để đáp ứng hạn chế, tôi xin đề xuất hướng nghiên cứu tiếp theo của luận văn là thực hiện trong môi trường thực, sử dụng đa dạng hơn nữa các thuật toán học tăng cường khác, sử dụng các thư viện mã nguồn mở như A2C, PPO, đây là các thuật toán hiện đại, cho phép thực hiện quá trình tính toán song song, giảm thời gian huấn luyện tác nhân. Các thuật toán này cũng đã được nhiều công trình nghiên cứu đề cập đến.