|  |
| --- |
| TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI  **VIỆN ĐIỆN TỬ - VIỄN THÔNG**  logo_128  BÁO CÁO  **THỰC TẬP TỐT NGHIỆP**  **Đề tài:**  **TRIỂN KHAI GIẢI PHÁP THÍCH ỨNG BĂNG THÔNG CHO VIDEO CBR**  Sinh viên thực hiện: NGUYỄN TRỌNG VINH  Lớp KT ĐTTT 05 – K59  Giảng viên hướng dẫn: Th.S. NGUYỄN THỊ KIM THOA    Hà Nội, 3-2019 |
| TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI  **VIỆN ĐIỆN TỬ - VIỄN THÔNG**  logo_128  BÁO CÁO  **THỰC TẬP TỐT NGHIỆP**  **Đề tài:**  **TRIỂN KHAI GIẢI PHÁP THÍCH ỨNG BĂNG THÔNG CHO VIDEO CBR**  Sinh viên thực hiện: NGUYỄN TRỌNG VINH  Lớp KT ĐTTT 05 – K59  Giảng viên hướng dẫn: Th.S. NGUYỄN THỊ KIM THOA  Cán bộ phản biện:  Hà Nội, 3-2019 |

# LỜI NÓI ĐẦU

Có thể thấy xem video là một hình thức giải trí không thể thiếu trong cuộc sống ngày nay. Các video có nội dung phong phú có sẵn trên khắp các trang mạng xã hội, đứng ở phía người dùng em luôn mong muốn được xem video có chất lượng tốt nhất, tuy nhiên có những lúc mạng Internet không cho phép làm được điều đó. Do học ngành Điện tử - Truyền thông có những kiến thức về mạng máy tính, đa phương tiện, lập trình…, cùng với hứng thú tìm hiểu về truyền thông đa phương tiện và được sự gợi ý đề tài từ Cô Nguyễn Thị Kim Thoa em đã tìm hiểu, nghiên cứu và triển khai thuật toán thích ứng chất lượng video khi streaming đồng thời nhiều video có tốc độ bit biến đổi nhằm cải thiện chất lượng cảm nhận của người xem. Người xem sẽ có cảm nhận về chất lượng video tốt hơn so với việc streaming video thông thường.

Em xin chân thành cảm ơn Th.S Nguyễn Thị Kim Thoa đã hướng dẫn em thực hiện, cùng các thành viên của phòng thí nghiệm ESRC đã giúp đỡ em hoàn thành khóa thực tập cuối khóa này.

# TÓM TẮT ĐỒ ÁN

Những năm gần đây, dịch vụ streaming video có nhu cầu ngày càng tăng đối với video có chất lượng cao. Nhờ có sự phát triển mạnh của các công nghệ mạng kết nối băng rộng, hiệu suất cao streaming video đang trở thành dịch vụ chính của Internet. Tuy nhiên thì việc cung cấp cho người dùng một video có chất lượng cao thì vẫn đang còn nhiều thách thức. Công nghệ streaming thích ứng ra đời giải quyết được được việc thích ứng chất lượng video CBR – video có tốc độ bit không đổi đối với các mạng có tốc độ bit khác nhau. Máy khách có tốc độ bit cao sẽ xem được video có chất lượng cao, tốc độ bit thấp sẽ xem video với chất lượng thấp. Tuy nhiên, đối với video VBR có tốc độ bit biến đổi trong suốt phiên streaming, sự biến động mạnh của tốc độ bit của video và thông lượng mạng ảnh hưởng trực tiếp đến cảm nhận người dùng là một thách thức lớn trong dịch vụ streaming video thích ứng. Các nghiên cứu về streaming video thích ứng đối với video có tốc độ bit biến đổi đang còn hạn chế. Vì vậy trong giới hạn của đồ án này, em đã tìm hiểu và nghiên cứu các mô hình, thông số ảnh hưởng đến chất lượng cảm nhận của người dùng từ đó triển khai thuật toán thích ứng chất lượng khi streaming đồng thời nhiều video VBR trên nền giao thức HTTP. Việc thích ứng chất lượng này nhằm cải thiện chất lượng cảm nhận của người dùng, nâng cao chất lượng dịch vụ streaming video đáp ứng nhu cầu xem video hiện nay.

# ABSTRACT

# MỤC LỤC

[LỜI NÓI ĐẦU 2](#_Toc5206402)

[TÓM TẮT ĐỒ ÁN 3](#_Toc5206403)

[ABSTRACT 4](#_Toc5206404)

[MỤC LỤC 5](#_Toc5206405)

[DANH SÁCH HÌNH VẼ 7](#_Toc5206406)

[DANH SÁCH BẢNG BIỂU 8](#_Toc5206407)

[DANH SÁCH TỪ VIẾT TẮT 9](#_Toc5206408)

[PHẦN MỞ ĐẦU 10](#_Toc5206409)

[CHƯƠNG 1: GIỚI THIỆU 12](#_Toc5206410)

[1.1. Cơ sở lý thuyết liên quan đến công nghệ streaming video 12](#_Toc5206411)

[1.1.1 Bộ giao thức TCP/IP 12](#_Toc5206412)

[1.1.2 Giao thức UDP 13](#_Toc5206413)

[1.1.3 Giao thức TCP 14](#_Toc5206414)

[1.1.4 Video 15](#_Toc5206415)

[1.2. Tổng quan về công nghệ streaming video 16](#_Toc5206416)

[1.2.1 Giải thích các thuật ngữ 16](#_Toc5206417)

[1.2.2 Công nghệ streaming video 17](#_Toc5206418)

[1.3. Công nghệ HTTP adaptive streaming video 20](#_Toc5206419)

[1.3.1 Giao thức HTTP 20](#_Toc5206420)

[1.3.2 Nguyên lý hoạt động của hệ thống HAS 21](#_Toc5206421)

[1.3.3 Các yếu tố ảnh hưởng tới QoE trong HAS 23](#_Toc5206422)

[1.4. Kết luận chương 23](#_Toc5206423)

[CHƯƠNG 2: TRIỂN KHAI GIẢI PHÁP THÍCH CHẤT LƯỢNG CHO VIDEO VBR 25](#_Toc5206424)

[2.1. Thích ứng tốc độ bit và mức chất lượng cho video 25](#_Toc5206425)

[2.2. Triển khai 28](#_Toc5206426)

[2.3. Kết quả thực hiện 29](#_Toc5206427)

[CHƯƠNG 3: KẾT LUẬN VÀ KẾ HOẠCH THỰC HIỆN ĐỒ ÁN TỐT NGHIỆP 30](#_Toc5206428)

[3.1. Kết luận 30](#_Toc5206429)

[3.2. Kế hoạch thực hiện Đồ án tốt nghiệp 30](#_Toc5206430)

[3.3. Đề cương Đồ án tốt nghiệp 31](#_Toc5206431)

[TÀI LIỆU THAM KHẢO 33](#_Toc5206432)

[PHỤ LỤC 34](#_Toc5206433)

[Phụ lục 1: Mã nguồn C của các thuật toán triển khai 34](#_Toc5206434)

# DANH SÁCH HÌNH VẼ

[Hình 1. 1 Tốc độ bit tại các phân đoạn một video VBR 16](#_Toc5206294)

[Hình 1. 2 Sơ đồ hệ thống HAS 21](#_Toc5206295)

[Hình 1. 3 Máy khách trong hệ thống HAS 22](#_Toc5206296)

[Hình 2. 1 Quan hệ giữ tốc độ bit đại diện R với d0 và tốc độ bit tức thời tại các phân đoạn. 26](#_Toc5238326)

[Hình 2. 2 Mô hình đường bao, dữ liệu tich luỹ và độ trễ ban đầu 27](#_Toc5238327)

[Hình 2. 3 Sơ đồ thuật toán tìm U max với mức băng thông R được cấp 28](#_Toc5238328)

[Hình 2. 4 Kết quả chạy trên video lam với mức băng thông được cấp 500 Kbps 29](#_Toc5238329)

[Hình 2. 5 Kết quả chạy trên video lam với mức băng thông được cấp 1000 Kbps 29](#_Toc5238330)

[Hình 2. 6 Kết quả chạy trên video lam với mức băng thông được cấp 2000 Kbps 30](#_Toc5238331)

# DANH SÁCH BẢNG BIỂU

[Bảng 1. 1 Các công nghệ streaming video 18](#_Toc5205336)

[Bảng 3. 1 Kế hoạch thực hiện Đồ án tốt nghiệp 30](#_Toc5205807)

# DANH SÁCH TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **Từ viết tắt** | **Tên tiếng Anh** | **Tên tiếng Việt** |
| CBR | Constant Bitrate | Tốc độ bit cố định |
| DASH | Dynamic Adaptive Streaming over HTTP | Streaming thích ứng động trên giao thức HTTP |
| HTTP | HyperText Transfer Protocol | Giao thức truyền siêu văn bản |
| IPTV | Internet Protocol Television | Truyền hình giao thức Internet |
| MPD | Media Presentation Description | Mô tả trình diễn phương tiện |
| QP | Quantization Parameter | Tham số lượng tử |
| RTCP | Real-time Control Protocol | Giao thức điều khiển thời gian thực |
| RTP | Real-time Transport Protocol | Giao thức truyền tải thời gian thực |
| TCP | Transport Control Protocol | Giao thức điều khiển truyền vận |
| UDP | User Datagram Protocol | Giao thức gói dữ liệu người dùng |
| VBR | Varible BitRate | Tốc độ bit biến đổi |

# PHẦN MỞ ĐẦU

Không lâu trước đây, khi nói đến xem TV nghĩa là ngồi xem TV trước phòng khách chờ đợi những trương trình chúng ta yêu thích được phát sóng vào các khung giờ cố định. Tuy nhiên ngày nay, với sự phát triển của Internet một hình thức dịch vụ video theo nhu cầu ra đời, người xem có thể xem các trương trình yêu thích tại bất kỳ thời gian nào nhờ dịch vụ streaming video. Theo một thống kê của Nielson Global, tại Việt Nam có đến 91% trong số những người được khảo sát cho biết họ có xem video theo nhu cầu trên cả ứng dụng và smart TV. Nhu cầu xem video theo sở thích ngày càng tăng lên mạnh mẽ. Trên nền tảng các mạng xã hội phổ biến, Youtube có đến năm tỉ video được xem mỗi ngày, thời gian video được xem trên Facebook hàng ngày lên đến 100 tỉ giờ . Với nhu cầu xem video rất lớn và không ngừng tăng cho thấy là streaming video đang trở thành một dịch vụ chính trên Internet ngày nay.

Do vậy, dịch vụ streaming video qua Internet cần được cải thiện để đem lại trải nghiệm tốt hơn cho người dùng. Thách thức chính của dịch vụ streaming video qua Internet là sự biến động thông lượng gây ra bởi các mạng không đồng nhất về băng thông ngày nay. Do sự biến động của thông lượng, video sẽ không thể được truyền đi với một tốc độ bit cố định. Do đó các chuẩn streaming được phát triển từ năm 2008 đều dựa vào công nghệ streaming thích ứng. Một trong những đặc trưng chính của công nghệ này là khả năng thay đổi tốc độ bit theo thông lượng hiện tại của mạng. Với các máy khách có tốc độ Internet khác nhau streaming thích ứng cho phép truyền video với các mức tốc độ bit khác nhau phù hợp với tốc độ bit của máy khách. Ví dụ một chiếc smart phone với một kết nối 3G có tốc độ thấp có thể xem được video với tốc độ bit thấp tương ứng, trong khi dùng máy tính với kết nối dây mạng có băng thông cao hơn có thể xem video với tốc độ bit cao. Hay nói cách khác streaming thích ứng cho phép phía máy có khách khả năng thích thông lượng kết nối nếu thông lượng biến đổi trong suốt một phiên truyền.

Những năm gần đây, kỹ thuật phổ biến cho streaming video qua mạng Internet là streaming thích ứng qua giao thức truyền siêu văn bản, viết tắt là HAS (Hyper Text Transfer Protocol Adaptive Streaming) . Hiện nay những nền tảng mạng xã hội lớn cung cấp video tới người dùng ứng dụng công nghệ HAS bao gồm Netflix, Youtube, Hulu và Amazone Instant Video.

Một số nghiên cứu liên quan đến chất lượng cảm nhận người dùng trong HAS , Tuy nhiên các nghiên cứu này vẫn chưa thực sự trong việc cải thiện chất lượng cảm nhận của người dùng đặc biệt là trong bối cảnh streaming video qua mạng di động với thông lượng mạng thường biến động mạnh theo thời gian. Có thể thấy tiềm năng chưa được khai thác hết của công nghệ HAS và nhu cầu mạnh mẽ của streaming video qua mạng di động những năm gần đây, giải pháp thích ứng chất lượng video trên nền HTTP để đối phó với sự biến động của thông lượng mạng, nâng cao chất lượng cảm nhận của người dùng là cần thiết.

Giới hạn trong nội dung của Thực tập tốt nghiệp, em tập trung vào nghiên cứu và triển khai các thuật toán thích ứng chất lượng một video VBR có tốc độ bit biến đổi với một băng thông cho trước nhằm nâng cao chất lượng cảm nhận của người dùng. Đây là tiền đề cho việc nghiên cứu và triển khai thuật toán phân bổ băng thông thích ứng chất lượng nhiều video VBR trên nền giao thức HTTP.

Nội dung báo cáo được trình bày theo các chương sau:

* **Chương 1: Giới thiệu** – Trình bày các kiến thức cơ sở, nền tảng liên quan đến công nghệ streaming video.
* **Chương 2: Nghiên cứu và triển khai giải pháp thích chất lượng cho video VBR** – Trình bày mô hình các thông số ảnh hưởng đến QoE và triển khai giải pháp thích ứng chất lượng cho một video VBR.
* **Chương 3: Kết luận và kế hoạch thực hiện Đồ án tốt nghiệp** – kết luận chung về đề tài, đánh giá kết quả đạt được và hướng phát triển, tối ưu trong tương lai, đưa ra đề cương chi tiết cho Đồ án tốt nghiệp.

# CHƯƠNG 1: GIỚI THIỆU

Trong chương này, em sẽ trình bày những lý thuyết cơ sở nền tảng liên quan đến công nghệ streaming video và công nghệ streaming thích ứng video trên nền tảng giao thức HTTP.

## Cơ sở lý thuyết liên quan đến công nghệ streaming video

### 1.1.1 Bộ giao thức TCP/IP

TCP/IP là một bộ các giao thức truyền thông được sử dụng trong Internet và các mạng máy tính. Tên gọi TCP/IP có nghĩa là giao thức TCP qua IP đây là hai giao thức nền tảng trong bộ giao thức này. TCP/IP bao gồm các giao thức thấp như IP, UDP, TCP. Bộ giao thức TCP/IP cung cấp truyền thông dữ liệu đầu cuối xác định rõ cách dữ liệu được đóng gói, xử lý, truyền, định tuyến và nhận. Các chức năng này được tổ chức thành bốn tầng trừu tượng, mỗi tầng giải quyết một vấn đề có liên quan đến việc truyền dữ liệu, và cung cấp cho các giao thức ở tầng trên một dịch vụ được định nghĩa rõ ràng dựa trên việc sử dụng dịch vụ từ các dịch vụ của tầng phía dưới. Từ thấp nhất là tầng liên kết dữ liệu (link layer) bao gồm các phương thức truyền dữ liệu có chức năng biến đổi dòng bit logic thành các tín hiệu vật lý phù hợp với đường truyền vật lý (ở bên phát) và điều chế giải điều chế, khôi phục tín hiệu (ở bên thu). Tầng internet (Internet layer) cung cấp liên kết giữa các mạng độc lập, tầng giao vận (transport layer) xử lý và giao tiếp giữa các máy chủ với nhau. Tầng ứng dụng (application layer) cung cấp trao đổi dữ liệu theo quy trình cho các ứng dụng.

TCP/IP có hai giao thức nền tảng chính ở tầng giao vận (transport layer), một là giao thức phi kết nối (connectionless protocol) và giao thức thứ hai là giao thức hướng kết nối (connection-oriented). Hai giao thức này bổ sung cho nhau. Giao thức phi kết nối là UDP, UDP không làm gì ngoài việc gửi các gói tin giữa các ứng dụng, cho phép các ứng dụng xây dựng giao thức của riêng chúng khi cần thiết. Giao thức hướng kết nối là TCP, TCP tạo các kết nối và là giao thức truyền tin tin cậy khi có cơ chế truyền lại, cùng với cơ chế điều khiển luồng, kiểm soát tắc nghẽn. Cả hai giao thức TCP và UDP đều hoạt động trên cơ sở của giao thức IP.

### 1.1.2 Giao thức UDP

UDP là một trong những bộ giao thức cốt lõi của giao thức TCP/IP. Giao thức này được thiết kế bởi David P. Reed năm 1980 và chính thức được định nghĩa trong RFC 768. Với UDP các ứng dụng máy tính có thể gửi các tin nhắn, trong trường hợp này được xem như là một datagram tới các máy khác trong giao thức mạng (Internet protocol network).

Giao thức UDP là giao thức đơn giản, phi liên kết và cung cấp dịch vụ trên tầng giao vận với tốc độ nhanh. UDP hỗ trợ kết nối một - nhiều và thường được sử dụng trong trong liên kết một nhiều bằng cách sử dụng các datagram multicast (cách thức truyền tin được gửi từ một điểm đến một tập hợp các điểm khác) và unicast (cách thức truyền tin được gửi từ một điểm tới một điểm khác).

Khi sử dụng UDP gói tin chỉ được gửi đến bên nhận, bên gửi sẽ không chờ đợi để xác nhận rằng bên nhận đã nhận được gói tin mà sẽ tiếp tục gửi các gói tin tiếp theo. Nếu bên nhận hủy bỏ một số gói tin, bên nhận sẽ không thể yêu cầu bên gửi gửi lại lần nữa. Không có gì để đảm bảo rằng bên nhận đã được tất cả các gói tin hay chưa nhưng bù vào đó các máy tính có thể giao tiếp với nhau một cách nhanh chóng hơn.

UDP thường được sử dụng khi có nhu cầu về tốc độ và độ chính xác của thông tin không yêu cầu quá cao. Ví dụ UDP thường được sử dụng cho các chương trình phát sóng trực tiếp và trò chơi trực tuyến.

**Nhược điểm UDP**

UDP không đảm bảo dữ liệu được truyền đến đích hay chưa. UDP không có cơ chế thiết lập kênh truyền nên bên gửi không biết được bên nhận đã nhận được gói tin hay chưa chưa nhận được. UDP không hỗ trợ các chỉ số phiên (session ID) để duy trì các liên kết kết giữa 2 máy do bản chất phi liên kết của nó. UDP không đảm bảo thứ tự các gói tin đến đích như đã được tạo ra ở nguồn.

Tính bảo mật không cao. Trong nhiều tổ chức, tường lửa và router cấm các gói tin UDP vì các hacker thường dò tìm các cổng UDP để biết được máy chủ đang chạy các dịch vụ mạng nào và thực hiện khai thác thông tin trái phép. UDP không có cơ chế kiểm soát luồng. Một ứng dụng UDP tồi có thể làm giảm băng thông của mạng.

**Các ưu điểm của UDP**

UDP không cần thiết lập liên kết. Không sử dụng các tín hiệu bắt tay nên có thể tránh được thời gian trễ. Do vậy DNS thường sử dụng giao thức UDP hơn là TCP.

UDP hỗ trợ các liên kết một – một, một – nhiều, trong khi TCP chỉ hỗ trợ liên kết một – một. Kích thước header 8 byte nhỏ hơn so với TCP là 20 byte giúp tiếc kiệm băng thông**.**

### 1.1.3 Giao thức TCP

UDP là một giao thức đơn giản và có một số công dụng rất quan trọng như tương tác giữa các máy khách và máy chủ và truyền thông đa phương tiện. Tuy nhiên hầu hết các ứng dụng sử dụng Internet thì độ tin cậy, tuần tự là cần thiết, UDP không thể đáp ứng đặc điểm đó vì vậy một giao thức khác được phát triển là TCP để bổ sung cho những tính năng thiếu sót của UDP.

TCP được định nghĩa chính thức trong tài liệu RFC 793 là một trong những bộ giao thức cốt lõi của bộ giao thức TCP/IP. TCP là giao thức hướng kết nối (connection-oriented) tức là khi muốn truyền nhận dữ liệu thì phải thiết lập kết nối trước. Sử dụng TCP các ứng dụng trên máy chủ được kết nối mạng có thể kết nối với nhau mà qua đó chúng có thể trao đổi dữ liệu là các gói tin và đảm bảo tính tin cậy, khả năng kiểm soát lỗi của dòng dữ liệu giữa các ứng dụng chạy trên các máy chủ đang kết nối với nhau thông qua giao thức IP. Các ứng dụng Internet đòi hỏi độ tin cậy cao của dữ liệu như là World Wide Web, email, quản trị từ xa, truyền dữ liệu đều dựa vào TCP.

TCP cung cấp dịch vụ truyền thông ở tầng trung gian giữa một trương trình ứng dụng và giao thức Internet chính là tầng giao vận của mô hình mạng Internet. Một ứng dụng không cần biết các cơ chế cụ thể để gửi dữ liệu qua một liên kết tới máy chủ khác ví dụ như xử lý phân mảnh gói tin để phù hợp với kích thước khung truyền dẫn. Tại tầng giao vận TCP xử lý tất cả các chi tiết bắt tay và truyền nhận giữa các máy chủ và tạo ra một kết nối mạng trừu tượng tới các ứng dụng thông qua giao diện socket. Socket là một đơn vị điểm đầu cuối được tạo ra bởi TCP để gửi và nhận dữ liệu. Mỗi socket có một địa chỉ IP của máy chủ và một số 16 bit dùng để định nghĩa port tương ứng với các dịch vụ mà máy chủ đang chạy. Như vậy một kết nối TCP phải được thiết lập rõ ràng giữa một socket trên một máy chủ và một socket trên một máy chủ khác.

TCP cung cấp một khả năng truyền dữ liệu an toàn giữa các máy chủ trong mạng với cơ chế thiết lập và huỷ bỏ kết nối. Cung cấp các chức năng kiểm tra chính xác dữ liệu khi đến đích và truyền lại dữ liệu khi có lỗi xảy ra. Sử dụng cơ chế báo nhận, đánh số thứ tự các gói dữ liệu và sắp xếp lại thứ tự dữ liệu tại bên nhận. Cung cấp khả năng đa kết nối cho các tiến trình khác nhau giữa hai máy chủ qua số hiệu cổng.

Tất cả các kết nối TCP đều là song công (full-duplex). Song công nghĩa là dữ liệu có thể được truyền và nhận cùng một thời điểm. Đồng thời TCP chỉ hỗ trợ kết nối một – một (point – to – point), mỗi kết nối chỉ có đúng hai điểm đầu cuối. TCP không hỗ trợ kết nối multicast hay broadcast.

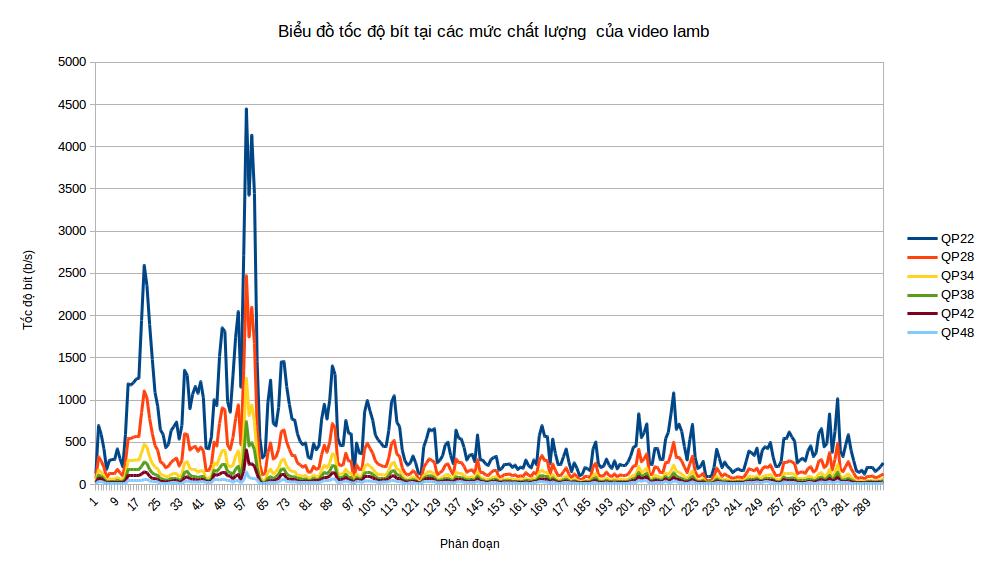
### 1.1.4 Video

Video là một đa phương tiện điện tử để ghi, sao chép, phát lại, truyền tải và hiển thị một cách trực quan của chuyển động. Video thường được lưu trữ và truyền đi sau khi mã hoá. Mục đích của việc mã hoá video là để giảm đi lượng dữ liệu dư thừa nhằm tiếc kiệm không gian lưu trữ cũng như băng thông khi truyền đi.

Một trong những tham số quan trọng quyết định đến chất lượng video được mã hoá chính là tham số lượng tử QP. Khi QP rất nhỏ mọi thông tin của video đều được giữ lại, khi tăng QP một số thông tin được loại bỏ để giảm dung lượng và tốc độ bit cho video. Vì vậy, QP được coi là tham số đặc trưng cho mức chất lượng của video có giá trị nằm trong khoảng từ 0 đến 51 . Với mỗi mức chất lượng cụ thể, các video được chia thành nhiều phân đoạn. Độ dài của mỗi phân đoạn này thường nằm trong khoảng từ 2 đến 10 giây .

**Video VBR**

Với cách mã hoá tốc độ bit biến đổi (VBR), mỗi mức chất lượng đặc trưng bởi tham số QP cố định, mỗi khung hình (hoặc phân đoạn) được mã hoá với kích thước khác nhau tuỳ thuộc vào nội dung video, làm cho chất lượng đầu ra không thay đổi trong khi tốc độ bit video thay đổi một cách tự nhiên. Điều này làm cho chất lượng của các khung hình được mã hoá sẽ tốt hơn khi sử dụng một tốc độ bit cố định. Hình 1.1 biểu diễn tốc độ bit của một video được mã hoá theo kiểu tốc độ bit biến đổi.



Hình 1. 1 Tốc độ bit tại các phân đoạn một video VBR

**Video CBR**

Một dạng mã hoá video khác được gọi là mã hoá với tốc độ bit cố định (CBR). Khác với VBR mã hoá video CBR với mỗi mức chất lượng thì tốc độ bit được giữ cố định trong suốt quá trình mã hoá. Mã hoá CBR rất hữu ích trong streaming các nội dung đa phương tiện trên kênh truyền có dung lượng hạn chế vì nó cho biết chính xác tốc độ bit thay vì chỉ biết tốc độ bit trung bình. Tuy nhiên CBR có nhược điểm là đối với những phân đoạn video nhiều cảnh tĩnh, ít chuyển động thì bị phân bố dư thừa dữ liệu. Và ngược lại những cảnh nhiều chuyển động lại không phân đủ dữ liệu dẫn đến chất lượng video đầu ra không ổn định.

## Tổng quan về công nghệ streaming video

### 1.2.1 Giải thích các thuật ngữ

Băng thông và thông lượng là hai khái niệm thường đi với nhau khi nói về một đường truyền vật lý, trong đó:

**Băng thông (bandwidth)** là khả năng truyền dữ liệu tối đa theo lý thuyết trong một đơn vị thời gian của một đường truyền vật lý, đơn vị bps.

**Thông lượng (throughput)** là lượng dữ liệu thực tế được gửi qua một kết nối đường truyền vật lý trong một đơn vị thời gian, đơn vị bps. Thông lượng luôn luôn nhỏ hơn hoặc bằng băng thông. Đối với video thông lượng được đo sau khi nhận được một phân đoạn video. Thông lượng được tính bằng tỉ số giữa kích thước dữ liệu của phân đoạn và khoảng thời gian tải phân đoạn đó. Khoảng thời gian tải phân đoạn đó được tính từ thời điểm máy khách gửi truy vấn cho đến khi nó nhận được byte cuối cùng của phân đoạn được truy vấn.

**Chất lượng cảm nhận của người dùng (Quality of Experience – QoE)** là phép đo về cảm nhận của người dùng như vui vẻ, hài lòng hay khó chịu khi sử dụng một dịch vụ và ở đây là chơi một video. Nó là kết quả của việc đáp ứng mong đợi của người đó đối với video. QoE tập trung vào toàn bộ các trải nghiệm của dịch vụ

Mức sử dụng bộ đệm: lượng video còn lại trong bộ đệm, thường được tính bằng đơn vị thời gian.

**Đa phương tiện (multimedia)** là nội dung mà sử dụng kết hợp các hình thức nội dung khác nhau như văn bản, âm thanh, hình ảnh, hoạt hình, video và các nội dung có tính tương tác. Đa phương tiện có thể được ghi lại và được phát, hiển thị, tương tác, hoặc được truy cập bởi các thiết bị xử lý thông tin như các thiết bị điện tử, máy tính. Tuy nhiên, đa phương tiện cũng có thể là một phần của một cuộc trình diễn trực tiếp. Các thiết bị đa phương tiện là các thiết bị truyền thông điện tử được sử dụng để lưu trữ và giúp người dùng trải nghiệm nội dung đa phương tiện.

**Truyền thông đa phương tiện (streaming media)** Là hình thức đa phương tiện được truyền và trình bày (chơi, phát, hiển thị) tới người dùng cuối một cách liên tục trong khi vẫn được truyền bởi một nhà cung cấp dịch vụ đa phương tiện. Ý nghĩa của động từ “stream” ở đây là nói về quy trình của việc truyền và nhận được đa phương tiện. Thuật ngữ này chỉ phương thức truyền và nhận của các loại đa phương tiện chứ không chỉ riêng mình các loại đa phương tiện hay hình thức tải toàn bộ nội dung của các tập tin đa phương tiện trước khi trình phát chúng. Người dùng cuối có thể dùng một chương trình để trình phát nội dung đa phương tiện trong khi một phần nội dung của đa phương tiện vẫn đang được truyền tải.

### 1.2.2 Công nghệ streaming video

**Đặc điểm và phân loại**

Sau khi chuẩn nén chuẩn quốc tế đầu tiên MPEG-1 và H261 ra đời vào đầu những năm 1990, công nghệ video bắt đầu được hình thàn và phát triển. Tại thời điểm đó, các video được số hoá, mã hoá và lưu như một file video tại máy chủ. Nếu một người dùng muốn xem một video thì cách duy nhất là tải toàn bộ tập tin video từ phía máy chủ về và sau đó dùng trình phát giải mã để chơi. Tuy nhiên phương pháp này có nhược điểm đó là người dùng phải đợi đến khi tải xong mới xem được video khoảng thời gian đợi này có thể là rất dài. Mặt khác khi tải tập tin video về người dùng sẽ tốn một phần không gian bộ nhớ để lưu trữ, dẫn đến vấn đề về tài nguyên máy tính. Một giải pháp cho những vấn đề này đó là streaming video. Trong một hệ thống ứng dụng streaming video, máy khách bắt đầu chơi video trong vòng một vài giây sau thời điểm nó bắt đầu nhận được dữ liệu từ máy chủ. Hay nói cách khác, người dùng sẽ không cần phải đợi đến khi tải hết toàn bộ nội dung của video từ máy chủ mà vẫn có thể xem video trong khi vẫn tải nội dung video từ máy chủ.

Streaming video trên Internet có thể được chia thành ba công nghệ chính như bảng 1.1 đó là unicast streaming, IP multicast, Application-layer multicast.

Bảng 1. 1 Các công nghệ streaming video

|  |  |
| --- | --- |
| **Công nghệ streaming video** | **Đặc điểm** |
| Unicast streaming | Thông thường sử dụng các bộ giao thức RTP/UDP hoặc HTTP/TCP để truyền video. Hiện tại HTTP adaptive streaming (HAS) đang được sử dụng phổ biến. |
| IP multicast | Có thể truyền video trên mạng truyền hình giao thức Internet. Tuy nhiên, IP multicast không được phát triển rộng vì cả hai lý do về kỹ thuật và kinh tế |
| Application-layer multicast | Sử dụng các giao thức riêng biệt (BitTorrent, …) để truyền video với các phương pháp Peer-to-Peer streaming. Tuy nhiên hình thức này gặp thách thức lớn trong việc bảo mật và chống các cuộc tấn công độc hại tới phía các máy khách. |

Ngoài cách phân loại theo công nghệ streaming như trên, streaming video còn được phân loại theo các đặc điểm kỹ thuật. Các kỹ thuật streaming video là rất đa dạng thường thì được phân loại theo năm yếu tố, đó là: yêu cầu về độ trễ, chế độ mã hoá video, các giao thức tầng giao vận, tính thích ứng và vị trí điều khiển thích ứng.

**Yêu cầu về độ trễ:** Dựa vào yêu cầu về độ trễ video streaming được chia thành ba loại: streaming tương tác, streaming trực tiếp và streaming theo yêu cầu. Đặc trưng của streaming tương tác và streaming trực tiếp là nội dung video được truyền tới người xem trong khi máy quay phía phát vẫn đang ghi nội dung tiếp theo của video. Hình thức streaming video này yêu cầu độ trễ rất thấp. Một ví dụ của streaming tương tác là hai người hoặc một nhóm sử dụng tính năng video call trên facebook để nói chuyện với nhau từ những vị trí địa lý khác nhau. Streaming trực tiếp là những trương trình trực tiếp như bóng đá U23 được phát trực tiếp trên website vtvgo của Đài truyền hình Việt Nam. Có thể thấy độ trễ trong các hình thức streaming này là rất quan trọng và ảnh hưởng trực tiếp đến người xem. Ngược lại với video theo nhu cầu, video được ghi trước và được lưu tại máy chủ nên nó không yêu cầu bất kỳ nào.

**Tốc độ bit mã hoá video:** Dựa vào tốc độ bit mã hoá video người ta chia ra làm hai loại là streaming video có tốc độ bit biến đổi (VBR) và streaming video có tốc độ bit không đổi (CBR). Việc dự đoán tốc độ bit cho một video dạng VBR là rất khó khăn vì vậy streaming video VBR đang còn nhiều thách thức. Với tốc độ bit cố định, streaming video CBR sẽ dễ dàng hơn so với video VBR.

**Giao thức truyền:** Dựa vào các đặc điểm của các giao thức tầng giao vận UDP và TCP, streaming video ban đầu sử dụng giao thức nền là UDP vì có lợi thế về tốc độ. Tuy nhiên sau này TCP lại trở thành sự lựa chọn phổ biến vì cung cấp nhiều lợi thế hơn về chi phí kinh tế cho hệ thống cũng như tính an toàn cao hơn so với sử dụng giao thức UDP.

**Tính thích ứng:** Có hai loại đó là streaming thích ứng và streaming không thích ứng. Streaming thích ứng mang lại nhiều lợi thế khi tốc độ bit được lựa chọn phù hợp với thông lượng mạng của người dùng. Trong khi streaming không thích ứng người dùng có khả năng sẽ bị gián đoạn khi xem vì thông lượng mạng biến đổi.

**Vị trí điều khiển thích ứng:** Trong hệ thống streaming thích ứng, khối điều khiển thích ứng có thể được đặt tại hai vị trí đó là tại máy chủ hoặc máy khách. Trước đây bộ thích ứng được đặt tại máy chủ, tuy nhiên do có quá nhiều máy khách cùng truy cập gây phức tạp và quá tải cho máy chủ. Do đó, hầu hết các dịch vụ streaming (trừ streaming tương tác) đều đặt khối thích ứng tại máy khách.

**Lịch sử phát triển từ streaming dữ liệu gói đến HAS**

Ban đầu khi mới ra đời streaming video trên mạng Internet sử dụng bộ giao thức RTP/RTCP/RTSP. RTP là chuẩn Internet cho truyền dữ liệu đa phương tiện thời gian thực. RTP mô tả hai giao thức con là giao thức truyền dữ liệu RTP và giao thức điều khiển RTCP. RCTP chỉ rõ thông tin điều khiển cho việc đồng bộ và chất lượng của các tham số dịch vụ có thể được gửi. RTP hoạt động dựa trên nền UDP có ưu điểm về tốc độ do giao thức có các cơ chế tối thiểu nên nó phù hợp với phân phối đa hướng, độ trễ truyền dữ liệu thấp, phù hợp với streaming tương tác. Tuy nhiên, việc truyền dữ liệu qua UDP là không đáng tin cậy.

Do những nhược điểm của giao thức RTP trên nền UDP, kỹ thuật streaming tải luỹ tiến ra đời sử dụng giao thức HTTP trên nền TCP. Tệp video được tải lên máy chủ web thông thường, máy khách chỉ đơn giản truy cập vào máy chủ và tải về bộ đệm cục bộ của nó các phân đoạn video sau đó hiển thị đoạn video được lưu trong bộ đệm trong khi dữ liệu vẫn tiếp tục được tải về bộ đệm. Nếu tốc độ trình chiếu vượt quá tốc độ tải xuống thì chương trình dừng trình chiếu cho đến khi dữ liệu được tải về. Nhược điểm của phương pháp này là không thích ứng tốc độ bit, mỗi máy khách được xem như nhau về mặt băng thông khả dụng. Máy khách chọn một mức chất lượng video và chỉ được chọn một lần ngay khi bắt đầu trong suốt phiên truyền. Điều này dẫn đến việc lãng phí băng thông khi mà thông lượng thay đổi lên cao hơn so với mức chất lượng được chọn hay gây nên sự gián đoạn khi xem video khi thông lượng thấp hơn mức chất lượng được chọn ban đầu. Giải pháp này là không thích ứng được với sự biến đổi thông lượng mạng vì vậy hạn chế này đã dẫn đến sự phát triển của công nghệ HAS – streaming thích ứng trên nền giao thức HTTP.

Trong hệ thống HAS, các máy chủ lưu nhiều phiên bản với các mức chất lượng khác nhau của một video và mỗi mức chất lượng, video được phân thành nhiều phân đoạn có độ dài bằng nhau. Sau khi chuẩn MPEG-DASH ra đời cho phép máy khách lựa chọn các mức chất lượng cho từng phân đoạn thay vì chọn một lần cho cả video. Trong phiên streaming một thuật toán đặt tại máy khách có nhiệm vụ thích ứng chất lượng video với điều kiện mạng, điều này sẽ giảm đáng kể các gián đoạn gây ra bởi biến đổi thông lượng mạng.

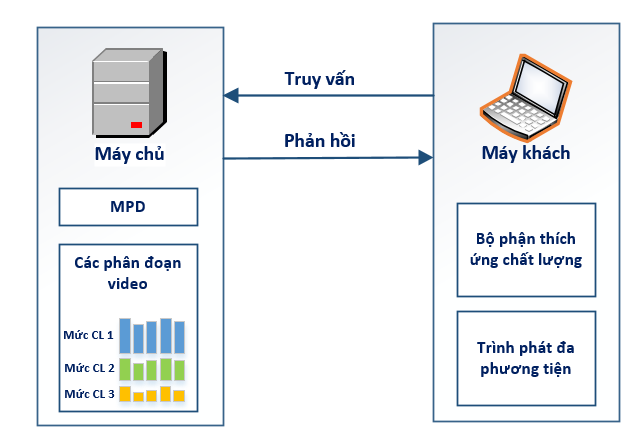
## Công nghệ HTTP adaptive streaming video

### 1.3.1 Giao thức HTTP

HTTP là giao thức được dùng để liên hệ thông tin giữa máy cung cấp dịch vụ (web server) và máy sử dụng dịch vụ (web client) trong mô hình máy khách/máy chủ dùng cho World Wide Web. HTTP là một giao thức thuộc tầng ứng dụng nằm trên tầng giao vận trong mô hình mạng TCP/IP. Chuẩn HTTP đang được sử dụng phổ biến hiện nay là HTTP/2 được ra đời và công bố trong tài liệu RFC 7540. HTTP hoạt động trên nền giao thức TCP, máy khách gửi truy vấn tới máy chủ và đợi dữ liệu trả về từ máy chủ. Quá trình này là liên tục trong suốt phiên kết nối. HTTP mang lại một số lợi ích rõ ràng cho các dịch vụ streaming như chỉ cần sử dụng máy chủ Web thay vì một máy chủ riêng chuyên biệt để điều khiển truyền.

### 1.3.2 Nguyên lý hoạt động của hệ thống HAS

Kiến trúc phổ biến của một hệ thống HAS bao gồm máy chủ, mạng truyền tải và máy khách được trình bày trên hình 1.2. Các phân đoạn video và tập mô tả trình diễn đa phương tiện (MPD) được lưu tại máy chủ và sẽ được truy vấn bởi máy khách. Máy khách khởi tạo một phiên streaming bằng việc tải một tệp MPD bao gồm một mô tả các phiên bản chất lượng khác nhau của video và các phân đoạn. Sau đó một bộ phận thích ứng được triển khai tại máy khách chịu trách nhiệm cho việc quyết định phân đoạn nào được yêu cầu, dựa vào tệp MPD và trạng thái hiện tại của mạng. Video sau đó sẽ được truyền tới máy khách thông qua một chuỗi truy vấn – phản hồi HTTP.



Hình 1. 2 Sơ đồ hệ thống HAS

**Các thành phần quan trọng trong hệ thống HAS**

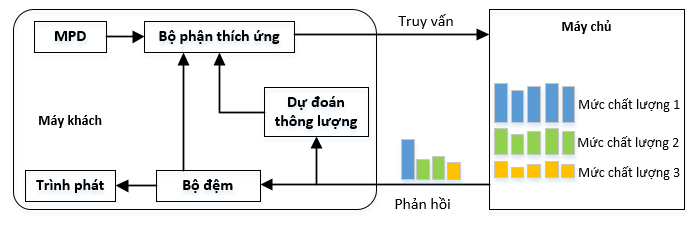
**Tệp MPD** là một tệp đuôi .xml chứa thông tin siêu dữ liệu bao gồm các thông tin như sau:

* Thông tin truyền tải chung như tên tệp, thông tin mật mã, truyền tải theo nhu cầu hay trực tiếp.
* Loại luồng như âm thanh, hình ảnh, phụ đề.
* Các mức chất lượng như thông tin về dạng mã hoá, tốc độ bit trung bình
* Chỉ số của các phân đoạn, đường dẫn tới các phân đoạn.
* Tốc độ đỉnh đối với video VBR

**Truyền video** trong hệ thống HAS, đồ án chỉ xét đến truyền các phân đoạn video tại tầng ứng dụng. Sau khi nhận được giữ liệu là tập MPD máy khách bắt đầu đưa ra một chuỗi các truy vấn HTTP để tải các phân đoạn video. Máy khách sẽ tuỳ vào thông lượng mạng mà chọn phân đoạn video có chất lượng phù hợp.

**Máy khách trong hệ thống HAS**

Hình 1.3 chỉ ra sơ đồ khối của máy khách trong hệ thống HAS. Tuỳ thuộc vào truy vấn của máy khách và phiên bản của giao thức HTTP, máy chủ sẽ phản hồi một hay nhiều phân đoạn với cùng một tốc độ bit. Tất các các phân đoạn được tải từ máy chủ sẽ được lưu ở bộ đệm máy khách trước khi chuyển tới trình chơi video.



Hình 1. 3 Máy khách trong hệ thống HAS

Bộ phận quan đóng vai trò quan trọng là bộ thích ứng, khi một phương pháp được phát triển để quyết định xem phân đoạn có chất lượng như thế nào sẽ được truy vấn. Một quyết định tốt cho một truy vấn cần dựa vào mức bộ đệm hiện tại, thông lượng được ước tính hiện tại và dữ liệu từ MPD. Tệp MPD là được nhận từ lúc bắt đầu phiên streaming, máy khách phải liên tục ức tính thông lượng và đo được mức sử dụng bộ đệm trong toàn bộ phiên streaming.

Tuỳ vào từng trường hợp cụ thể sẽ có những cách dự đoán thông lượng khác nhau. Một số cách dự đoán thông lượng điển hình như đo thông lượng của một phân đoạn hiện tại sau đó sử dụng một mô hình tuyến tính để dự đoán thông lượng cho phân đoạn kế tiếp. Ngoài ra có thể đo thông lượng của nhiều phân đoạn gần với phân đoạn hiện tại nhất và tính trung bình để dự đoán cho thông lượng kế tiếp.

Bộ đệm trong máy khách đóng vai trò quan trọng trong việc đối phó với biến động của thông lượng mạng. Bộ đệm giúp cho việc streaming video không bị gián đoạn. Bộ đệm với kích thước lớn, khả năng xảy ra gián đoạn sẽ nhỏ hơn bộ đệm có kích thước nhỏ. Kích thước và mức sử dụng bộ đệm được tính theo đơn vị thời gian. Khi bắt đầu một phiên streaming máy khách sẽ nạp vào bộ đệm một lượng dữ liệu nhất định trước khi giải mã và hiển thị dữ liệu ra màn hình. Đây là giai đoạn nạp bộ đệm ban đầu, sau đó máy khách chuyển qua giai đoạn streaming nhận dữ liệu và hiển thị đồng thời. Trong giai đoạn này nếu mức sử dụng bộ đệm đủ lớn người xem sẽ không bị gián đoạn bởi thông lượng mạng biến đổi gây ra.

### 1.3.3 Các yếu tố ảnh hưởng tới QoE trong HAS

Các yếu tố chính ảnh hưởng tới QoE được điều khiển bởi máy khách trong HAS bao gồm: Độ trễ nạp bộ đệm ban đầu, sự gián đoạn, chất lượng video, độ trễ trực tiếp (streaming trực tiếp) .

Độ trễ nạp ban đầu luôn tồn tại trong streaming video vì máy khách cần tải một lượng dữ liệu nhất định vào bộ đệm trước khi hiển thị video trên màn hình.

Sự gián đoạn là sự kiện đang xem video bị dừng do mức dữ liệu trong bộ đệm rỗng. Sự gián đoạn video gồm hai yếu tố là tần suất gián đoạn và khoảng thời gian gián đoạn đều làm ảnh hưởng tới QoE. Sự gián đoạn này xảy ra trong khi người dùng đang xem video và nó có ảnh hưởng nghiêm trọng hơn là độ trễ ban đầu.

Chất lượng cảm nhận được đánh giá thông qua độ nét của hình ảnh mà người dùng nhận được, được thể hiện bởi tốc độ bit (đối với video CBR) hoặc tham số lượng tử QP (đối với video VBR). Tốc độ bit càng cao QP càng thấp thì chất lượng được cung cấp càng tốt.

Độ trễ trực tiếp là thời gian từ lúc hình ảnh được chụp bởi camara tới lúc hiển thị trên màn hình đây là yếu tố quan trọng trong streaming video trực tiếp. Các thành phần trễ chính bao gồm chuẩn bị nội dung, trễ phân đoạn, thời gian tải phân đoạn, thời gian nạp vào bộ đệm và thời gian mã hoá.

## Kết luận chương

Trong chương I em đã trình bày các cơ sở lý thuyết nền tảng liên quan và công nghệ streaming video nói chung, đặc biệt là công nghệ HAS nói riêng. Sau đó là các yếu tố tác động đến QoE trong HAS. Đây sẽ là kiến thức để em có thể thực hiện các chương tiếp theo.

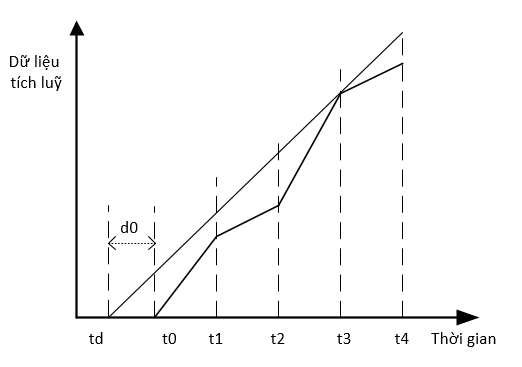
# CHƯƠNG 2: TRIỂN KHAI GIẢI PHÁP THÍCH CHẤT LƯỢNG CHO VIDEO VBR

Trong chương này, em sẽ trình bày về mô hình toán học các tham số ảnh hưởng đến QoE của người dùng từ đó đưa ra giải pháp và triển khai thích ứng chất lượng tốt nhất cho một video VBR với một mức băng thông trong nhiều mức băng thông được cấp cho video đó.

## Thích ứng tốc độ bit và mức chất lượng cho video

Một mức chất lượng video thường được đại diện thông qua tốc độ bit đại diện (kí hiệu là R) của video đó. Tốc độ bit đại diện càng cao thì chất lượng video càng cao và ngược lại. Đối với video CBR thì tốc độ bit đại diện chính là tốc độ bit dùng để mã hoá video đó. Với video VBR tốc độ bit đại diện liên quan đến hai yếu tố là độ trễ ban đầu và thời điểm máy khách bắt đầu hiển thị video . Độ trễ ban đầu được tính là khoảng thời gian mà máy khách bắt đầu nhận video cho tới thời điểm máy khách bắt đầu hiển thị video.

Tốc độ bit đại diện cho một mức chất lượng của video được tính dựa vào độ trễ ban đầu d0 và tốc độ bit tức thời tại các phân đoạn của video. Hình 2.1 chỉ ra tốc độ bit R chính là hệ số góc đường tiếp tuyến của đường gấp khúc có hệ số góc là các tốc độ bit tức thời của các phân đoạn video tại một mức chất lượng.



Hình 2. 1 Quan hệ giữ tốc độ bit đại diện R với d0 và tốc độ bit tức thời tại các phân đoạn.

Có thể thấy khi d0 tăng thì tốc độ bit đại diện của video tại mức chất lượng sẽ giảm. Như vậy nếu người xem chấp nhận đợi một khoảng thời gian chờ dài thì sẽ xem được video có chất lượng tốt hơn. Tuy nhiên nếu d0 quá dài thì QoE sẽ bị ảnh hưởng vì vậy cần phải có một mô hình thoả hiệp giữa chất lượng cảm nhận và độ trễ ban đầu. Một mô hình được đề xuất trong được thực hiện bằng cách thay thế một số phân đoạn có tốc độ bit cao xuống phân đoạn tương ứng có tốc độ bit thấp hơn. Việc thay thế này sẽ dẫn đến hệ số góc của đường tiếp tuyến sẽ bị bé lại tức là hạ được tốc độ bit đại diện cũng như giảm được độ trễ ban đầu sao cho chất lượng video giảm không đáng kể.

Mô hình cụ thể như sau: một video C có M mức chất lượng , mỗi mức chất lượng được đặc trưng bởi tham số lượng tử và gồm N phân đoạn với tốc độ bit . Các mức chất lượng đã được sắp xếp từ cao xuống thấp . Cần thích ứng video này với một mức băng thông được cấp R và một ràng buộc DT là giới hạn của độ trễ ban đầu.

Các tác giả trong đã định nghĩa một hàm tại mỗi phân đoạn nếu tốc độ bit của nó lớn hơn thì được thay thế bởi phân đoạn tương ứng ở mức chất lượng thấp hơn gần nhất mà có tốc độ bit nhỏ hơn .

(2. 1)

Ngưỡng tốc độ bit sẽ được lấy rời rạc. Với mỗi hàm ta tìm được một mức chất lượng thích ứng có chất lượng và độ trễ ban đầu là từ một tốc độ bit đại diện R được cấp cho video. Bài toán đặt ra là tìm sao cho lợi ích U của người dùng là lớn nhất. Lợi ích U được định nghĩa như là hàm bù trừ giữa chất lượng video và độ trễ ban đầu:

(2.2)

Việc tính U được đề xuất thông qua mô hình tuyến tính tại [12] như sau:

(2.3)

Với là hàm của độ trễ ban đầu d0 thể hiện sự ảnh hưởng của thời gian chờ đợi đến QoE. là hàm của tham số lượng tử QP thể hiện sự ảnh hưởng của chất lượng đến QoE. Trọng số a và b thể hiện mức độ ảnh hưởng của độ trễ ban đầu so với chất lượng video thường được lấy tương ứng là 0.2 và 0.8.

Mức chất lượng thích ứng Q\* của video được tính như sau:

(2.4)

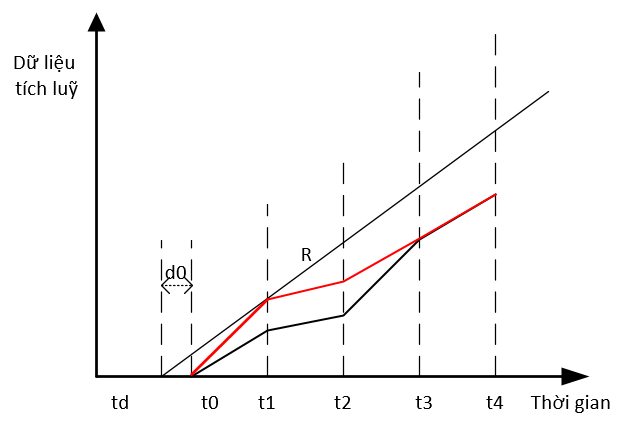
Với là tham số lượng tử cho phân đoạn thứ j của mức chất lượng thích ứng. Công thức để tính được đề xuất trong [13] được tính như sau:

(2.5)

Để tính độ trê ban đầu d0 cần dựa vào khái niệm đường bao. Đường bao E của một luồng video tại mỗi thời điểm t là dữ liệu tới hạn có thể được nhận bởi máy khách từ thời điểm bắt đầu nhận cho đến thời điểm t.

(2.6)

Với là lượng dữ liệu tích luỹ tính từ thời điểm ban đầu nhận được video đến thời điểm t. Tại hình 2.2 đường màu đỏ chính là đường bao E. Đường thẳng với tốc độ bit đại diện R tiếp tuyến với đường bao sẽ thu được độ trễ ban đầu d0.



Hình 2. 2 Mô hình đường bao, dữ liệu tich luỹ và độ trễ ban đầu

Từ hình 2.2 ta có

(2.7)

Công thức 2.7 chỉ ra rằng với thời gian chờ ban đầu là d0 thì phiên streaming sẽ không bao giờ bị gián đoạn cho dù máy khách hiển thị video tại bất kỳ thời điểm tp nào.

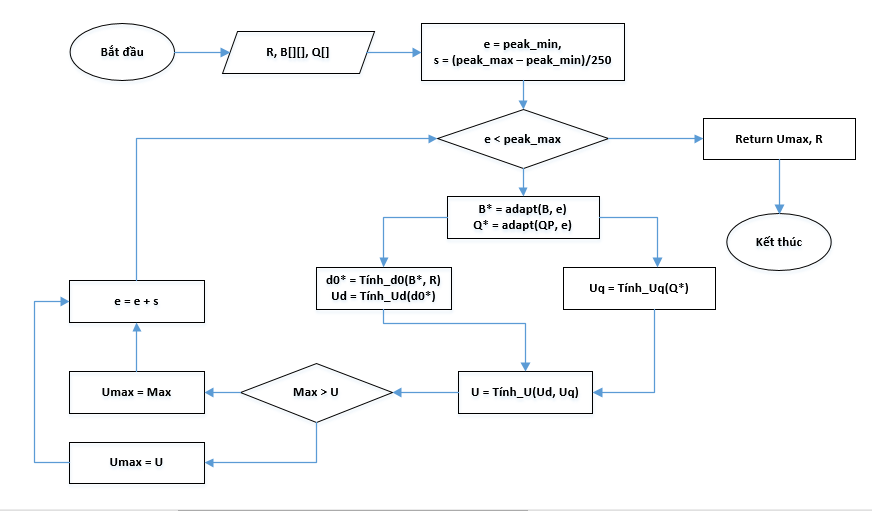
Để tính sử dụng mô hình đề xuất trong [13] như sau :

(2.8)

Như vậy với mức băng thông được cấp cho video là R, tìm được một mức chất lượng video mạng lại giá trị lợi ích U cao nhất, trong giới hạn độ trễ .

## Triển khai

Thuật toán được triển khai trên ngôn ngữ C và sử dụng trình biên dịch gcc. Hình 2. 3 thể hiện sơ đồ thuật toán tổng quát.



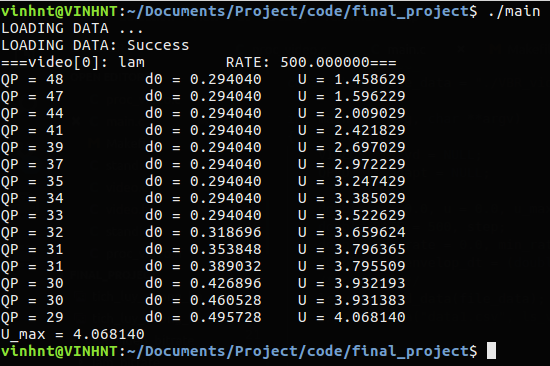
Hình 2. 3 Sơ đồ thuật toán tìm U max với mức băng thông R được cấp

Độ trễ ban đầu d0\* được tính bằng cách lấy giao điểm của đường tiếp tuyến có hệ số góc R với đường dữ liệu đường bao như hình 2.2.

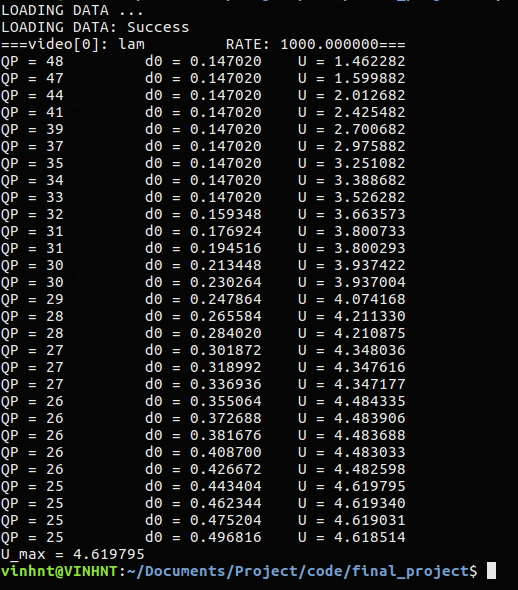
Với giá trị được cho chạy từ giá trị tốc độ bit nhỏ nhất của phân đoạn trong video đến tốc độ bit lớn nhất của một phân đoạn trong video đó. Như vậy với mỗi băng thông được cấp R ta tìm được một mức chất lượng Q\* sau đó tính giá trị U cho mỗi mức chất lượng thích ứng từ đó tìm giá trị lợi ích U lớn nhất. Kết quả này sẽ là đầu vào cho giải pháp phân bổ băng thông cho streaming nhiều video VBR.

## Kết quả thực hiện

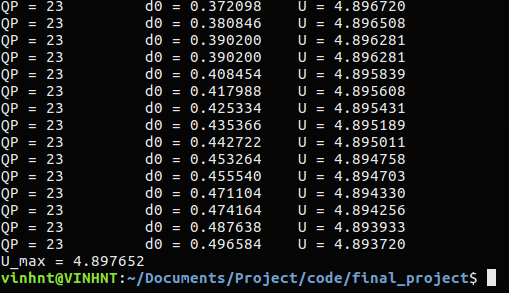
Kết quả được thực hiện trên video lamp với các mức băng thông được cấp là 500 Kbps, 1000 Kbps, 2000 Kbps.



Hình 2. 4 Kết quả chạy trên video lam với mức băng thông được cấp 500 Kbps



Hình 2. 5 Kết quả chạy trên video lam với mức băng thông được cấp 1000 Kbps



Hình 2. 6 Kết quả chạy trên video lam với mức băng thông được cấp 2000 Kbps

Có thể thấy rằng khi băng thông được cấp tăng lên thì mức chất lượng video được cải thiện đáng kể dẫn đến U tăng.

# 

# CHƯƠNG 3: KẾT LUẬN VÀ KẾ HOẠCH THỰC HIỆN ĐỒ ÁN TỐT NGHIỆP

## 3.1. Kết luận

Trong quá trình thực hiện thực tập tốt nghiệp em đã tìm hiểu được các cơ sở lý thuyết nền tảng về công nghệ streaming và đặc biệt là công nghệ HAS. Từ đó em đã thực hiện tìm hiểu và triển khai mô hình để thích ứng chất lượng để đạt được giá trị lợi ích lớn nhất đối với người xem cho một video VBR với các mức băng thông được cấp cho video. Đây là nền tảng để em có thể thực hiện triển khai phân bổ băng thông cho streaming nhiều video VBR với một mức băng thông tổng được cấp.

Em xin chân thành gửi lời cảm ơn đến Th.s Nguyễn Thị Kim Thoa đã định hướng và giúp đỡ em hoàn thành đợt thực tập tốt nghiệp này.

## 3.2. Kế hoạch thực hiện Đồ án tốt nghiệp

Bảng 3. 1 Kế hoạch thực hiện Đồ án tốt nghiệp

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên công việc** | **Ngày bắt đầu** | **Ngày kết thúc** |
| 1 | Tìm hiểu và nghiên cứu các đặc trưng ảnh hưởng đến QoE trong streaming video | 18/03/2018 | 24/03/2019 |
| 2 | Tìm kiếm tài liệu về các mô hình phân bổ băng thông cho video VBR trong hệ thống HAS hiện có | 25/03/2018 | 07/04/2019 |
| 3 | Áp dụng mô hình thích ứng chất lượng cho một video VBR để tìm ra các mức băng thông cho lợi ích người dùng tốt nhất. | 08/04/2018 | 29/04/2019 |
| 4 | Xây dựng mô hình phân bổ băng thông cho streaming nhiều video VBR với một mức băng thông tổng để đạt được lợi ích người dùng lớn nhất. | 30/03/2018 | 30/04/2019 |
| 5 | Triển khai các mô hình và đánh giá so sánh kết quả của mô hình triển khai với các mô hình có sẵn. | 30/04/2018 | 20/05/2019 |

## 3.3. Đề cương Đồ án tốt nghiệp

Từ kế hoạch thực hiện Đồ án tốt nghiệp, em xin đưa ra đề cương Đồ án tốt nghiệp của em như sau:

CHƯƠNG 1 : GIỚI THIỆU

1.1. Cơ sở lý thuyết nền tảng

1.1.1. Bộ giao thức TCP/IP

1.1.2. Giao thức UDP

1.1.3. Giao thức TCP

1.1.4. Video

1.2. Tổng quan về công nghệ streaming video

1.2.1. Giải thích các thuật ngữ

1.2.2. Lịch sử phát triển công nghệ streaming video

1.2.3. Đặc điểm và phân loại

1.3. Tổng quan về công nghệ HAS

1.3.1. Giao thức HTTP

1.3.2. Nguyên lý hoạt động của HAS

1.3.3 Các yếu tố ảnh hưởng đến QoE trong streaming video

1.4. Kết luận chương

CHƯƠNG 2: XÂY DỰNG VÀ TRIỂN KHAI THUẬT TOÁN THÍCH ỨNG CHẤT LƯỢNG CHO VIDEO VBR

2.1. Các mô hình hiện có

2.2. Xây dựng thuật toán triển khai mô hình

2.3. Triển khai mô hình và kết quả triển khai

2.4. Kết luận chương

CHƯƠNG 3: XÂY DỰNG VÀ TRIỂN KHAI PHÂN BỔ BĂNG THÔNG CHO NHIỀU VIDEO VBR

3.1. Giải pháp triển khai

3.2. Xây dựng thuật toán triển khai

3.3. Đánh giá và so sánh kết quả triển khai với mô hình có sẵn

3.4. Kết luận chương

CHƯƠNG 4: KẾT LUẬN.

# TÀI LIỆU THAM KHẢO

[1]. <https://www.renderforest.com/blog/video-marketing-statistics>, truy nhập cuối cùng ngày 26/3/2019

[2]. Begen, A. C., T. Akgul, and M. Baugher (2011, Mar). Watching video over the web: Part i: Streaming protocols. *IEEE Internet Computing 15(2), 54-63.*

[3]. Joseph, V. and G. de Veciana (2014, April). Nova: QoE-driven optimization of dash-based video delivery in networks. In *IEEE INFOCOM 2014 – IEEE conference on Computer Communications,* pp. 82-90.

[4]. Song, W. and D. W. Tjondronegoro (2014, April). Acceptability-based QoE models for mobile video. *IEEE Transactions on Multimedia 16*(3), 738-750

[5]. Golikeri, A., P. Nasiopoulos, and Z. Wang (2007, 10). Robust digital video water-marking scheme for h.264 advanced video coding standard. 16, 043008.

[6]. Thang, T. C., Q.-D. Ho, J. W. Kang, and A. T. Pham (2012, Feb). Adaptive streaming over http with dynamic resource prediction. In *Proc. Fourth International Conference on Communications and Electronics (ICCE 2012),* pp. 130-135.

[7] Le, H. T., T. Vu, N. P. Ngoc, A. T. Pham, and T. C. Thang (2017). Seamless mobile video streaming over http/2 with gradual quality transitions. *IEICE Transactions on Communications (accepted).*

[8]. Le, H. T. (2017). *Quality Improvement for HTTP Adaptive Streaming over Mobile Information Systems 2016, Article ID 292085.*

[10]. Cong Thang, T. and Y. M. Ro (2003, 05). Pratical estimation techniqes of trafic specification for vbr video services. *Proceedings of SPIE – The International Society for Optical Engineering 5022.*

[11]. Nguyen, D. V., D. M. Nguyen, H. T. Tran, N. P. Ngoc, A. T. Pham, and T. C. Thang (2015, Jan). Quality-delay tradeoff optimization in multi-bitrate adaptive streaming. In *2015 IEEE International Conference on Consumer Electronics (ICCE),* pp. 66-67.

[12]. T. C. Thang, J. Y. Lee, J. W. K. S. J. B. S. J. S. T. P. (2011, Jan). Bandwidth information for dash. ISO/IEC JTC1/SC29/WG11 m19324, Daegu.

[13]. Nguyen, D. V., H. T. Le, A. T. Pham, T. C. Thang, J. Y. Lee, and K. Yun (2013) Adaptive home surveillance system using htttp streaming. In *2013 International Joint Conference on Awareness Science and Technology Ubi-Media Computing (iCAST 2013 UMEDIA 2013),* pp. 579-584

# PHỤ LỤC

## Phụ lục 1: Mã nguồn C của các thuật toán triển khai

Tệp main.c

|  |
| --- |
| #include <stdio.h>  #include "video.h"  #include "standio.h"  #include "proc\_video.h"  #define \_DEBUG\_  const char \*file\_data = "./VBR\_video\_bitrate/";  int main(int arg, char \*\*argv)  {  VIDEOS \*ls\_vd = NULL;  QPVD \*ls\_adapt = NULL;  double e = 0.0, u = 0.0;  double RATE = 1000;  double max\_rate = 0.0;  double \*ls\_envelop\_dt = (double \*) malloc(sizeof(double) \* num\_seg);  /\*Load data\*/  ls\_vd = load\_data(file\_data);  //print\_data("data1.csv", ls\_vd, num\_vd, num\_seg, num\_qp);    #ifndef \_DEBUG\_  for (int i = 0; i < num\_vd; i++) {  printf("==========video[%d]: %s===========\n", i, ls\_vd[i].vd\_name);  max\_rate = find\_rate\_max(ls\_vd[i].ls\_qpvd[0].ls\_rate);  e = max\_rate/100;  printf("max\_rate %f\n", max\_rate);  while (e < max\_rate) {  ls\_adapt = pick\_adapt\_R(e, ls\_vd[i].ls\_qpvd);  u = cal\_U(ls\_adapt, RATE);  if (u > 0) printf("e = %f\nU = %f\n", e, u);  e += max\_rate/100;  }  e = 0.0;  }  //printf("Epxilone = %f\n", e);  #endif  #ifdef \_DEBUG\_  double \*ls\_data\_tich\_luy = NULL;  double \*env\_data = NULL;  QPVD \*adapt\_data = NULL;  printf("DEBUG MODE\n");  //ls\_data\_tich\_luy = data\_tich\_luy(ls\_vd[0].ls\_qpvd[0].ls\_rate);  adapt\_data = pick\_adapt\_R(500,ls\_vd[0].ls\_qpvd);  print\_list\_rate("./data\_print/adapt\_data500.csv", adapt\_data->ls\_rate);  //printf("QP adapt = %d", adapt\_data->qp);  ls\_data\_tich\_luy = data\_tich\_luy(adapt\_data->ls\_rate);  print\_list\_rate("./data\_print/acc\_data500.csv", ls\_data\_tich\_luy);  env\_data = envelop\_vd(adapt\_data->ls\_rate);  printf("Max rate: %f\n", find\_rate\_max(adapt\_data->ls\_rate));  //d0 = cal\_d0(env\_data, RATE);  print\_list\_rate("./data\_print/envelop\_data500.csv", env\_data);  #endif  return 0;  } |

Tệp video.h

|  |
| --- |
| #ifndef \_VIDEO\_H  #define \_VIDEO\_H  #define MAXCHAR 100  #define NUM\_VIDEO 5  #define MAX\_NUM\_OF\_SEGMENTS 294  #define MAX\_PATH\_LEN 256  #define NUM\_QP 6  #define INIT\_QP 1  static unsigned int QUANTIZIATION\_PARAMS[] = {22, 28, 34, 38, 42, 48};  static char \*VIDEO\_NAME[10] = {"lam", "sony", "star", "terminate", "tokyo"};  struct QP\_VIDEO {  double \*ls\_rate;  unsigned int qp;  };  #define QPVD struct QP\_VIDEO  struct LIST\_QP\_VIDEO {  QPVD \*ls\_qpvd;  char \*vd\_name;  };  static unsigned int num\_qp = NUM\_QP;  static unsigned int num\_vd = NUM\_VIDEO;  static unsigned int num\_seg = MAX\_NUM\_OF\_SEGMENTS;  #define VIDEOS struct LIST\_QP\_VIDEO  /\* Ham khoi tao list video \*/  QPVD \*init\_qpvd();  /\* Ham khoi tao video co nhieu qp \*/  VIDEOS \*init\_lst\_vd();  /\* Find max bitrate of video segments in set of QP \*/  double find\_rate\_max(double \*ls\_rate);  #endif |

Tệp video.c

|  |
| --- |
| #include "video.h"  #include <stdio.h>  #include <stdlib.h>  QPVD \*init\_qpvd()  {  QPVD \*qpvd = (QPVD \*) malloc(sizeof(QPVD));  qpvd->ls\_rate = (double \*) malloc(sizeof(double) \* num\_seg);  qpvd->qp = 0;    perror("Malloc");  return qpvd;  }  VIDEOS \*init\_lst\_vd()  {  VIDEOS \*vd\_lsqp = (VIDEOS \*) malloc(sizeof(VIDEOS));  vd\_lsqp->vd\_name = (char \*)malloc(sizeof(char) \* 128);  vd\_lsqp->ls\_qpvd = (QPVD \*)malloc(sizeof(QPVD) \* num\_qp);  return vd\_lsqp;  }  double find\_rate\_max(double \*ls\_rate)  {  double seg\_max\_rate = 0.0;  printf("Finding maximum rate of segment\n");  for (int i = 0; i < num\_seg; i++) {    seg\_max\_rate = (seg\_max\_rate > ls\_rate[i]) ? seg\_max\_rate : ls\_rate[i];  //printf("%d\t %f\t %f\n",i, ls\_rate[i], seg\_max\_rate);  }  printf("Find maximum rate done!\n");  return seg\_max\_rate;  } |

Tệp standio.h

|  |
| --- |
| #ifndef \_STANDIO\_  #define \_STANDIO\_  #include "video.h"  double str2num(char \*str);  VIDEOS \*load\_data(const char \*data\_dir);  void print\_data(const char \*filename, VIDEOS \*ls\_vd,  unsigned int num\_vd, unsigned int num\_seg, unsigned int num\_qp);  void print\_list\_rate(const char \*file\_name, double \*ls\_rate);  #endif |

Tệp standio.c

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  #include "standio.h"  #include "video.h"  VIDEOS \*load\_data(const char \*data\_dir)  {  VIDEOS \*ls\_vd = NULL;  unsigned int k = 0;  FILE \*fd = NULL;  char \*file\_name = NULL;  char str[MAXCHAR];  ls\_vd =(VIDEOS \*) malloc(num\_vd \* sizeof(VIDEOS));  for (int i = 0; i < num\_vd; i++) {  ls\_vd[i].ls\_qpvd = (QPVD \*) malloc(sizeof(QPVD) \* num\_qp);  for (int j = 0; j < num\_qp; j++) {  ls\_vd[i].ls\_qpvd[j].ls\_rate = (double \*) malloc(sizeof(double) \* num\_seg);  }  }  printf("LOADING DATA ...\n");  for (int i = 0; i < num\_vd; i++) {  /\* Get data for one video \*/  for (int j = 0; j < num\_qp; j++) {  file\_name = malloc(MAX\_PATH\_LEN \* sizeof(char));  snprintf(file\_name, MAX\_PATH\_LEN,  "%s%s\_bitrate/%sqp%d.txt",  data\_dir, VIDEO\_NAME[i], VIDEO\_NAME[i], QUANTIZIATION\_PARAMS[j]);  //printf("Get data from file directory: %s\n", file\_name);  fd = fopen(file\_name, "r");  if (fd == NULL) {  perror("Could not open file name");  exit(EXIT\_FAILURE);  }  ls\_vd[i].vd\_name = VIDEO\_NAME[i];  ls\_vd[i].ls\_qpvd[j].qp = QUANTIZIATION\_PARAMS[j];  /\* Get data for segment \*/  while (fgets(str, MAXCHAR, fd)) {  ls\_vd[i].ls\_qpvd[j].ls\_rate[k] = str2num(str);  //if (k < 5) printf("%f\t", ls\_vd[i].ls\_seg[k].ls\_rate[j]);  k += 1;  }  k = 0;  free(file\_name);  fclose(fd);  }  }  perror("LOADING DATA");  return ls\_vd;  }  void print\_data(const char \*filename, VIDEOS \*ls\_vd,  unsigned int n\_vd, unsigned int n\_seg, unsigned int n\_qp)  {  FILE \*fd = fopen(filename, "w");  if (fd == NULL) {  perror("Could not open file name");  exit(EXIT\_FAILURE);  }  //printf("numvd: %d, numseg: %d, numqp: %d\n", num\_vd, num\_seg, num\_qp);  for (int i = 0; i < num\_vd; i++) {  fprintf(fd, "%s\n", ls\_vd[i].vd\_name);  for (int j = 0; j < num\_qp; j++) {  fprintf(fd, "QP%d", ls\_vd[i].ls\_qpvd[j].qp);  for (int k = 0; k < num\_seg; k++) {  fprintf(fd, ",%f", ls\_vd[i].ls\_qpvd[j].ls\_rate[k]);  }  fprintf(fd, "\n");  }  fprintf(fd, "\n");  }  fclose(fd);  printf("PRINT DATA DONE!\n");  }  double str2num(char \*str)  {  int i;  double val = 0.0 , power;  for (i = 0; (str[i] >= '0' && str[i] <= '9'); i++) {  val = val \* 10.0 + (str[i] - '0');  }  if (str[i] == '.') {  i++;  }  for (power = 1.0 ; (str[i] >= '0' && str[i] <= '9'); i++) {  val = val \* 10.0 + (str[i] - '0');  power \*= 10;  }  return val/power;  }  void print\_list\_rate(const char \*file\_name, double \*ls\_rate)  {  FILE \*fd = NULL;  fd = fopen(file\_name, "w");  for (int i = 0; i < num\_seg; i++) {  printf("%f\t", ls\_rate[i]);  fprintf(fd, ",%f", ls\_rate[i]);  if (((i+1)%5) == 0) printf("\n");  }  printf("\n");  fclose(fd);  } |

Tệp proc\_video.h

|  |
| --- |
| #ifndef \_PROC\_VIDEO\_H  #define \_PROC\_VIDEO\_H  #include "video.h"  #include "standio.h"  QPVD \*pick\_adapt\_R(double e, QPVD \*ls\_qpvd);  double \*data\_tich\_luy(double \*ls\_rate);  void print\_data\_tich\_luy(const char \*file\_name, VIDEOS \*ls\_vd);  double \*envelop\_vd(double \*ls\_dt\_tl);  double cal\_d0(double \*ls\_ev\_data, double rate\_r);  double cal\_U(QPVD \*adapt\_vd, double rate);  #endif |

Tệp proc\_video.c

|  |
| --- |
| #include <stdlib.h>  #include <stdio.h>  #include <errno.h>  #include <math.h>  #include "proc\_video.h"  #include "standio.h"  QPVD \*pick\_adapt\_R(double e, QPVD \*ls\_qpvd)  {  QPVD \*vd\_adapt = (QPVD \*) malloc(sizeof(QPVD));  vd\_adapt->ls\_rate = (double \*) malloc(sizeof(double) \* num\_seg);  vd\_adapt->qp = 0;  double r\_adapt = 0.0;  unsigned int qp\_adapt = 0, sum\_qp\_adapt = 0;  for (int i = 0; i < num\_seg; i++) {  for (int j = 0; j < num\_qp; j++) {  if (ls\_qpvd[num\_qp - 1].ls\_rate[i] > e) {  r\_adapt = ls\_qpvd[num\_qp - 1].ls\_rate[i];  qp\_adapt = ls\_qpvd[num\_qp - 1].qp;  } else if (ls\_qpvd[j].ls\_rate[i] < e)  {  if (r\_adapt < ls\_qpvd[j].ls\_rate[i]) {  r\_adapt = ls\_qpvd[j].ls\_rate[i];  qp\_adapt = ls\_qpvd[j].qp;  } else {  r\_adapt = r\_adapt;  qp\_adapt = qp\_adapt;  }  }  }  vd\_adapt->ls\_rate[i] = r\_adapt;  sum\_qp\_adapt += qp\_adapt;  r\_adapt = 0.0;  }  vd\_adapt->qp = sum\_qp\_adapt/num\_seg;  //print\_list\_rate("./data\_print/test\_adapt.csv", vd\_adapt->ls\_rate);  return vd\_adapt;  }  double \*data\_tich\_luy(double \*ls\_rate)  {  double sum = 0.0;  double \*ls\_dt\_tl = (double \*) malloc(sizeof(double) \* num\_seg);  for (int i = 0; i < num\_seg; i++) {  sum += ls\_rate[i];  ls\_dt\_tl[i] = sum;  }  return ls\_dt\_tl;  }  void print\_data\_tich\_luy(const char \*file\_name, VIDEOS \*ls\_vd)  {  double sum = 0.0;  VIDEOS \*ls\_vd\_tl = (VIDEOS \*) malloc(sizeof(VIDEOS) \* num\_vd);  FILE \*fd = NULL;  fd = fopen(file\_name, "w");  if (fd) {  perror("Could not open file to write");  exit(EXIT\_FAILURE);  }  for (int i = 0; i < num\_vd; i++) {  ls\_vd\_tl[i].ls\_qpvd = (QPVD \*) malloc(sizeof(QPVD) \* num\_qp);  for (int j = 0; j < num\_qp; j++) {  ls\_vd\_tl[i].ls\_qpvd[j].ls\_rate = (double \*) malloc(sizeof(double) \* num\_seg);  }  }    for (int i = 0; i < num\_vd; i++) {  fprintf(fd, "%s", ls\_vd\_tl[i].vd\_name);  for (int j = 0; j < num\_qp; j++) {  for (int k = 0; k < num\_seg; k++) {  fprintf(fd, ",%f", ls\_vd\_tl[i].ls\_qpvd[j].ls\_rate[k]);  }  fprintf(fd, "\n");  }  }  fclose(fd);  printf("Ghi du lieu tich luy vao file %s\n", file\_name);  }  double \*envelop\_vd(double \*ls\_dt)  {  double \*env\_vd = NULL, \*dt\_tmp = NULL;  unsigned int k = 0;  double R\_max = 0.0;  unsigned int index = num\_seg;  dt\_tmp = (double \*) malloc(sizeof(double) \* num\_seg);  env\_vd = (double \*) malloc(sizeof(double) \* num\_seg);  //print\_list\_rate("./data\_print/data\_adapt.csv", ls\_dt);  for (int i = 0; i < num\_seg; i++) {  dt\_tmp[i] = 0.0;  }  for (int i = 0; i < num\_seg; i++) {  k = 0;  while ((k < index) && (index > 0)) {  /\* D[k] = D[k] + R[i+k]  i = 0, k = 0: D[0] = D[0] + R[0] (D[0] = R[0])  i = 0, k = 1: D[1] = D[1] + R[1] (D[1] = R[1])  i = 0, ....  i = 1, k = 0: D[0] = D[0] + R[1] (D[0] = R[0] + R[1])  i = 1, k = 1: D[1] = D[0] + R[2] (D[1] = R[1] + R[2])  i = 1, k = 292: D[292] = R[292] + R[293]  \*/  //printf("%d\t %d\n", i, k);  dt\_tmp[k] = dt\_tmp[k] + ls\_dt[i + k];  #ifdef \_DEBUG\_  if (i < 2) printf("dtmp[%d] = %f\t data[%d] = %f\n", k, dt\_tmp[k], k, ls\_dt[k]);  #endif  k++;  // printf("%d\n", k);  }  for (int j = 0; j < index; j++) {  R\_max = (R\_max > dt\_tmp[j]) ? R\_max : dt\_tmp[j];  //printf("R max = %f\n", R\_max);  }  index -= 1;  env\_vd[i] = R\_max;  R\_max = 0.0;  }  //printf("Envelop data done!\n");  return env\_vd;  }  double cal\_d0(double \*ls\_ev\_data, double rate\_r)  {  double delta\_max = 0.0, delta = 0.0;  for (int i = 0; i < num\_seg; i++) {  delta = ls\_ev\_data[i] - rate\_r \* i;  delta\_max = (delta\_max > delta) ? delta\_max : delta;  }  //printf("------------delta max = %f\n", delta\_max);  if ((delta\_max >= 0) && (delta\_max < 0.5 \* rate\_r)) {  //printf("delta max = %f\n", delta\_max);  return delta\_max/rate\_r;  }  return -1;  }  double cal\_U(QPVD \*adapt\_vd, double rate)  {  double Ud = 0.0, Uq = 0.0, d0 = 0.0;  double \*ls\_data\_tl = NULL, \*env\_vd = NULL;  env\_vd = envelop\_vd(adapt\_vd->ls\_rate);  d0 = cal\_d0(env\_vd, rate);  if (d0 < 0)  return -1;  printf("QP = %d\t\td0 = %f\n", adapt\_vd->qp, d0);  Uq = -0.172 \* adapt\_vd->qp + 9.249;  Ud = -0.862 \* log((d0 + 6.718)) + 5;  return 0.8 \* Uq + 0.2 \* Ud;  } |

Tệp Makefile

|  |
| --- |
| CC = gcc  CFLAGS = -Wall  .PHONY = main clean  SRCS = main.c standio.c video.c proc\_video.c  OBJS = main.o standio.o video.o proc\_video.o  DEPS = standio.h video.h proc\_video.h  # tu file .c tao file .o  # tu file .o tao ra file main to execute  main: $(OBJS)      @echo "==========BUILD============"      $(CC) -o $@ $^ $(CFLAGS) -lm  build/%.o: %c $(DEPS)      @echo "======CREATING OBJECT======"      $(CC) -c -o $@ $< $(CFLAGS)  clean:      @echo "========REMOVE=========="      rm -r \*.o main |