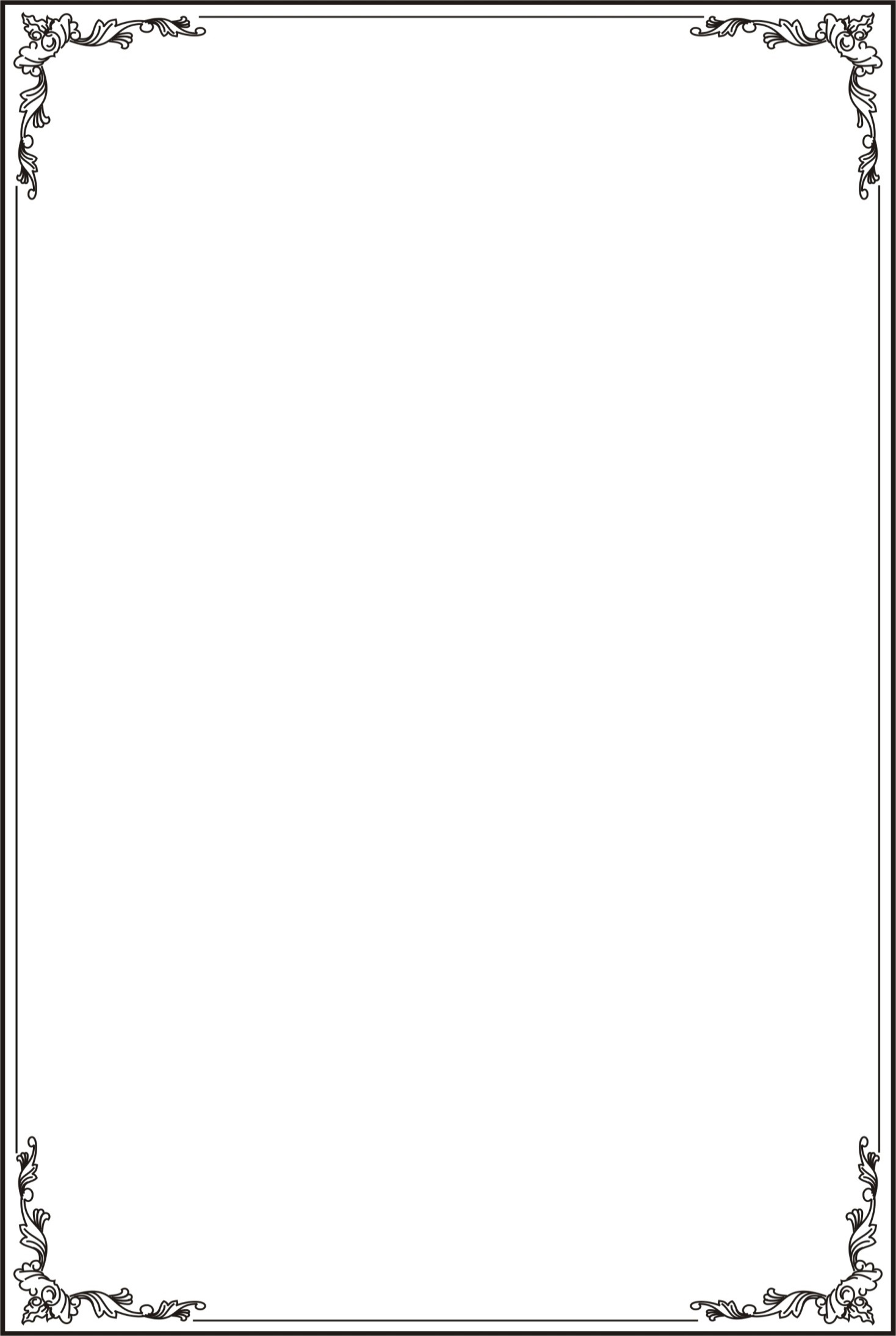
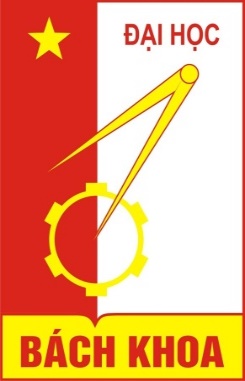
****

**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**VIỆN ĐIỆN TỬ - VIỄN THÔNG**

-----🙞🙜🕮🙞🙜-----



**BÁO CÁO**

**MÔN: ĐA PHƯƠNG TIỆN NÂNG CAO**

**Đề tài: Truyền thông đa phương tiện thời gian thực sử dụng VLC**

|  |  |  |
| --- | --- | --- |
| Giảng viên hướng dẫn | : TS. Phạm Văn Tến | |
| Sinh viên thực hiện | : Đỗ Sơn Tùng | 20145083 |
|  | : Bùi Đức Tường | 20145168 |
|  | : Lê Trung Kiên | 20142401 |
|  | : Trần Ngọc Anh | 20140236 |
| Lớp | : KSTN - ĐTVT K59 | |

Hà Nội – 23/11/2018

**LỜI NÓI ĐẦU**

Trong những năm trở lại đây, việc truyền tải và lưu trữ video càng trở nên phỏ biến hơn do yếu tố giá thành các sản phẩm liên quan như máy quay video HD,TV màn hình lớn, các thiết bị lưu trữ… đang giảm rất nhanh. Ngoài ra, cấc bộ xử lý mới ngày càng có tốc độ cao hơn rất nhiều, cùng với tốc độ internet được cải thiện đáng kể đã giúp việc truyền tải một bộ phim nhanh chóng và xem chúng được mượt mà hơn. Nhưng chừng đó là chư đủ, do nhu cầu cuộc sống luôn ngày càng lơn. Trước đây, việc thưởng thức các bộ phim ở độ phân giải 480p rồi đến mHD 720p đã là thỏa mã, thì giờ đây các bộ phim với độ phân giải Full HD 1080p đã tràn ngập, Nhu cầu đó ngày càng thể hiện mạnh mẽ hơn khi vài năm gần đây các tivi với công nghệ hình ảnh 4K đã làm cho người xem như đắm chìm vào thế giới thực. Các thiết bị lưu trữ và đường truyền internet với áp lực lớn hơn đã không kịp đáp ứng. Vì thế nhu cầu về một chuẩn nén video mới hiệu quả hơn nhằm giảm tái áp lực cho các hệ thống ngày càng bức thiết hơn. Và HEVC được ra đời nhằm đáp ứng các nhu cầu trên. HEVC là một chuẩn tiên tiến ra đời chính thức vào năm 2013. Nó tạo ra sự đột phá, cho phép nén video một cách tốt hơn đồng thời cải thiện được chất lượng so với các chuẩn trước đó. HEVC tăng gấp đôi tỉ lệ nén dữ liệu so với H.264 tại cùng một chất lượng video. Các tập tin nén video sẽ là một nửa kích thước so với H.264. Ngoài ra H.265 cũng có thể hỗ trợ video lên đến 8K UHD. Với thời đại internet bùng nổ, đa phương tiện mà chủ yếu là video đang chiếm song, đây là một công nghệ hữu ích giảm chi phí, giảm băng thông và tang chất lượng. Chúng em đã chọn việc nghiên cứu và đánh giá chuẩn HEVC với đề tài : “Streaming video với chuẩn video HEVC”.

[[https://scontent.fhan5-3.fna.fbcdn.net/v/t1.0-1/p50x50/24068340_789151747935144_2241566330471383134_n.jpg?_nc_cat=111&_nc_ht=scontent.fhan5-3.fna&oh=8e29ee5c5caab0309fa2f64478a8e523&oe=5C71AB2C](https://www.facebook.com/pham.vankhanh.5437)](https://www.facebook.com/pham.vankhanh.5437)

[[https://scontent.fhan5-3.fna.fbcdn.net/v/t1.0-1/p50x50/24068340_789151747935144_2241566330471383134_n.jpg?_nc_cat=111&_nc_ht=scontent.fhan5-3.fna&oh=8e29ee5c5caab0309fa2f64478a8e523&oe=5C71AB2C](https://www.facebook.com/pham.vankhanh.5437)](https://www.facebook.com/pham.vankhanh.5437)

**MỤC LỤC**

[DANH SÁCH HÌNH VẼ 3](#_Toc531325104)

[Chương 1. Giới thiệu về kĩ thuật mã hóa video H265 và chương trình truyền thông đa phương tiện VLC 4](#_Toc531325105)

[1.1. Kĩ thuật mã hóa video H265 4](#_Toc531325106)

[1.1.1. Tổng quan về H265/HEVC 4](#_Toc531325107)

[1.1.2. Cách thức hoạt động của H265 6](#_Toc531325108)

[1.2. Chương trình truyền thông đa phương tiện VLC 11](#_Toc531325109)

[1.2.1. Giới thiệu tổng quát 11](#_Toc531325110)

[1.2.2. Tiến hành tải mã nguồn và cài đặt 12](#_Toc531325111)

[Chương 2. Thiết lập một hệ thống tạo luồng tín hiệu đa phương tiện cho tiếng và hình (video streaming) giữa hai máy tính (sử dụng chuẩn H265) 14](#_Toc531325112)

[2.1. Stream video trên một máy tính sử dụng VLC 14](#_Toc531325113)

[2.2. Xem video streaming trên một máy tính khác 17](#_Toc531325114)

[Chương 3. Phát triển thêm tiện ích truyền văn bản nhanh giữa hai máy tính nói trên trong cùng giao diện GUI. 20](#_Toc531325115)

[3.1. Giới thiệu về Socket 20](#_Toc531325116)

[3.1.1. Khái niệm 20](#_Toc531325117)

[3.1.2. Khởi tạo socket 20](#_Toc531325118)

[3.1.3. Truyền tin nhắn văn bản nhanh giữa hai hay nhiều máy 23](#_Toc531325119)

[Chương 4. KẾT LUẬN 25](#_Toc531325120)

[TÀI LIỆU THAM KHẢO 26](#_Toc531325121)

# DANH SÁCH HÌNH VẼ

[Hình 1.1 Cách mã hóa video trong H.264 và H.265​ 5](#_Toc531325126)

[Hình 1.2 Cấu trúc của một bộ mã hóa và giải mã H.256/HEVC 7](#_Toc531325127)

[Hình 1.3 Quy trình làm việc của hệ thống mã hóa video 7](#_Toc531325128)

[Hình 1.4 Dự đoán định hướng trong H.264 và H.265 8](#_Toc531325129)

[Hình 1.5 Phân chia các Block thành các TU 9](#_Toc531325130)

[Hình 1.6 So sánh H.264/AVC và H.265/HEVC 10](#_Toc531325131)

[Hình 1.7 So sánh thời gian tải video giữa H265 và H264 khi cùng tốc độ mạng 11](#_Toc531325132)

[Hình 2.1 Khởi tạo server stream video 14](#_Toc531325133)

[Hình 2.2 Chọn video muốn stream 15](#_Toc531325134)

[Hình 2.3 Chọn giao thức stream video 16](#_Toc531325135)

[Hình 2.4 Định dạng video muốn stream 16](#_Toc531325136)

[Hình 2.5 Video đang được stream 17](#_Toc531325137)

[Hình 2.6 Nhập đường dẫn đến tập tin ảo 18](#_Toc531325138)

[Hình 2.7 Video được hiển thị trên máy khác 19](#_Toc531325139)

# Giới thiệu về kĩ thuật mã hóa video H265 và chương trình truyền thông đa phương tiện VLC

## Kĩ thuật mã hóa video H265

### Tổng quan về H265/HEVC

H.265/HEVC là một chuẩn nén video hoàn toàn mới với tên gọi đầy đủ là High Efficiency Video Coding (HEVC) được Ủy ban Viễn thông Quốc tế ITU-T thông qua và bắt đầu được các nhà phát triển đưa vào sản phẩm thương mại. H.265 hứa hẹn mang lại khả năng nén cao gấp đôi so với người tiền nhiệm H.264/AVC (Advanced Video Coding) hiện đang được dùng phổ biến khi chiếm tới 80% các nội dung video lưu trữ hiện nay.

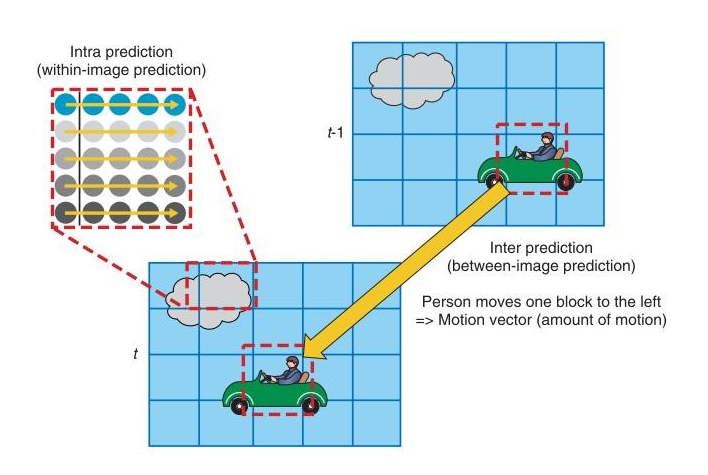
HEVC có tỷ lệ nén dữ liệu gấp đôi so với H.264/MPEG-4 AVC ở cùng một mức độ chất lượng video. Nói cách khác nó có thể được sử dụng để cung cấp chất lượng hình ảnh được cải thiện đáng kể khi xét cùng một bitrate . Nó có khả năng hỗ trợ 8K UHD (Ultra high definition television) và độ phân giải lên đến 8192x4320 (4320P).

H.265/HEVC cũng mang lại những cải tiến về âm thanh, không gian, màu sắc và quan trọng nhất nó có khả năng giám sát một phạm vi hoạt động nâng cao.  
Với việc cho chất lượng hình ảnh tốt như H.264/AVC, đồng thời nó lại có khả năng nén tốt hơn rất nhiều so với H.264/AVC. Điều này được cho là rất quan trọng nếu chúng ta muốn trải nghiệm [4K / Ultra HD](https://translate.googleusercontent.com/translate_c?depth=1&hl=en&ie=UTF8&prev=_t&rurl=translate.google.com.vn&sl=en&tl=vi&u=http://www.cnet.com/news/ultra-hd-4k-tv-cheat-sheet/&usg=ALkJrhiRxRhPwLloYM9q3sgm72p1WY16sg) một cách dễ dàng nhất. Khả năng nén cao còn giúp giảm băng thông cần thiết để truyền tải phim, giảm dung lượng lưu trữ từ đó giúp chúng ta phải trả ít chí phí hơn cho băng thông internet cũng như chi phí bỏ ra để mua thiết bị lưu trữ. Ưu điểm vượt trội này cũng sẽ là cú hích cho thị trường thiết bị nghe nhìn 4K/UHD.  
Về mặt lý thuyết H.265/HEVC được đánh giá là hiệu quả hơn H.264/AVC khoảng 30-50%. Ở độ phân giải càng cao, thì hiệu suất nén của H.265/HEVC càng cao. H.265/HEVC cơ bản có cấu trúc tương tự như các tiêu chuẩn trước đó như MPEG-2 và H.264/AVC.

Cả hai chuẩn mã hóa này đều dựa trên các kỹ thuật mã hóa video:

* Phân vùng hình ảnh bằng các macroblock và sau đó lại phân vùng tiếp trong các khối này.
* Giảm không gian dư thừa bằng cách sử dụng các kỹ thuật nén trong khung hình nội bộ.
* Giảm sự dư thừa thời gian bằng sử dụng kỹ thuật nén liên khung (dự đoán chuyển động và bù đắp chuyển động).
* Nén các dữ liệu dư thừa sử dụng các biến đổi toán học và lượng tử hóa.
* Giảm dư thừa trong truyền vector chuyển động và tín hiệu sử dụng mã hóa entropy.

Entropy thông tin là một khái niệm mở rộng của entropy trong nhiệt động lực học và cơ học thống kê sang lĩnh vực lý thuyết thông tin. Entropy thông tin mô tả mức độ hỗn loạn trong một tín hiệu lấy từ một sự kiện ngẫu nhiên. Nói cách khác, entropy cũng chỉ ra có bao nhiêu thông tin trong tín hiệu, với thông tin là các phần không hỗn loạn ngẫu nhiên của tín hiệu.  
 Một chuỗi các khung hình từ video nguồn sẽ được mã hoá hoặc nén lại bởi bộ mã hoá H.265/HEVC, kết quả là tạo ra một bitstream video đã được nén. Bitstream đã nén này có thể được lưu trữ hoặc truyền đi. Sau đó một bộ giải mã video sẽ giải nén bitstream này thành một chuỗi các khung hình để có thể trình chiếu video.



Hình . Cách mã hóa video trong H.264 và H.265​

Cùng với việc kế thừa từ H.264/AVC thì H.265/HEVC cũng có nhiều cải tiến mới như:

* Phân vùng (Partition) linh hoạt hơn, với kích cỡ từ lớn đến nhỏ.
* Linh hoạt hơn trong các chế độ dự báo (prediction modes) và chuyển đổi kích thước khối (transform block).
* Bộ lọc nội suy và bộ lọc khử khối (interpolation and deblocking) tinh vi hơn.
* Các chế độ dò tìm/dự báo và phát tín hiệu (prediction and signalling of modes), chuyển động của các vector (motion vectors) phức tạp hơn.
* Được trang bị tính năng hỗ trợ xử lý song song hiệu quả.

Kết quả là HEVC cung cấp một tiêu chuẩn mã hoá video có khả năng nén tốt hơn giúp tăng khả năng xử lý hình ảnh.

### Cách thức hoạt động của H265

H.265/HEVC cũng như các chuẩn mã hóa video khác đều có những logic trong quá trình tìm ra phương thức nén sao cho hiệu quả nhất. Các video khi quay bằng các máy quay HD chuyên nghiệp có dung lượng rất lớn do dữ liệu ở dạng thô và vì vậy để truyền tải các nội dung này đến người xem là rất khó khăn. Để thuận tiện trong truyền tải và lưu trữ, các video này được nén bằng các phương thức nén khác nhau nhằm giảm dung lượng xuống mức tối đa.

Video nguồn bao gồm một chuỗi các khung hình video sẽ được mã hóa hay nén bởi một bộ mã hóa video để tạo ra bitstream video nén. Bitstream nén được lưu trữ hoặc truyền đi. Một bộ giải mã video giải nén bitstream để tạo ra một chuỗi các khung giải mã.

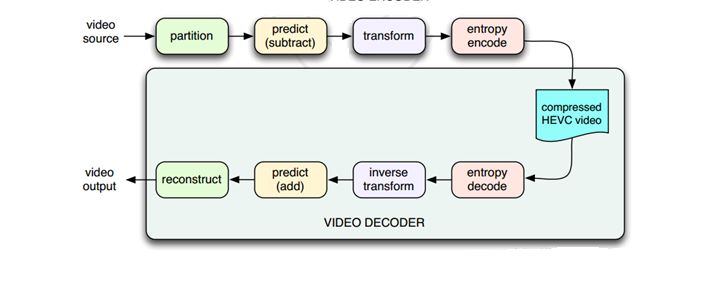
Các bước thực hiện bởi một bộ mã hóa video bao gồm:

* Phân vùng mỗi bức ảnh thành nhiều đơn vị.
* Dự đoán mỗi đơn vị bằng cách sử dụng liên hoặc nội dự đoán, và trừ các dự đoán từ các đơn vị.
* Chuyển đổi lượng tử còn sót lại (sự khác biệt giữa các đơn vị hình ảnh ban đầu và dự đoán).
* Entropy mã hóa các biến đổi đầu ra, thông tin dự báo, chế độ thông tin và các tiêu đề.

Một bộ giải mã video sẽ đảo ngược các bước :

* Entropy giải mã và trích xuất các yếu tố của chuỗi mã.
* Thay đổi tỉ lệ và đảo ngược giai đoạn chuyển đổi.
* Dự đoán từng đơn vị và thêm các dự đoán cho đầu ra của nghịch đảo.
* Xây dựng lại một hình ảnh video giải mã.

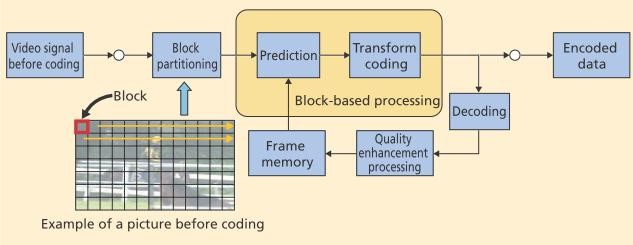
H.265/HEVC hỗ trợ phân vùng của một chuỗi video liên tục rất linh hoạt. Mỗi khung hình video hoặc hình ảnh đã được mã hóa sẽ phân chia thành các Titles và Slices sau đó tiếp tục được phân chia thành Coding Tree Units (CTUs). CTU là đơn vị cơ bản của mã hóa, tương tự như các macroblock trong các tiêu chuẩn trước đó, và kích thước có thể lên đến 64×64 pixel. Một CTU có thể được chia thành các vùng ô vuông được gọi là Coding Units (CUS) sử dụng cấu trúc Quadtree.



Hình 1.2 Cấu trúc của một bộ mã hóa và giải mã H.256/HEVC

**a, Dự đoán (Prediction)**

Dự đoán là kỹ thuật chính tạo nên các phương pháp mã hóa video H.256/HEVC.Video đầu vào được chia ra thành các khối n x n pixel và được mã hóa trong các đơn vị khối này. Các khối đầu vào được mã hóa thành bit stream (dãy tín hiệu nhị phân bao gồm 0 và 1) thông qua các quá trình bao gồm dự đoán và trực giao biến đổi.

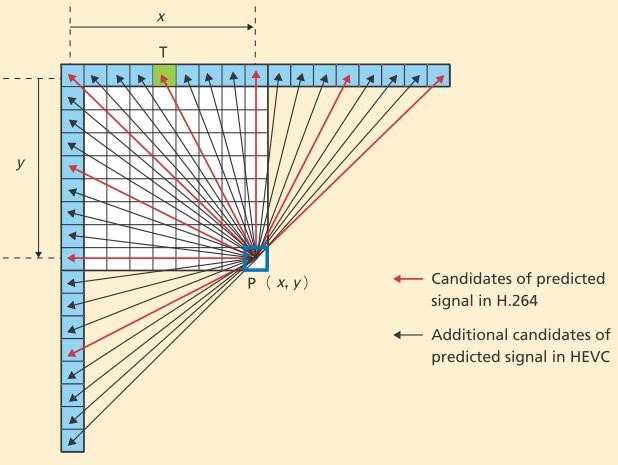


Hình 1.3 Quy trình làm việc của hệ thống mã hóa video

​

**b, Quy trình làm việc của hệ thống mã hóa video**

Trong quá trình mã hóa khung hình tại thời điểm t, nếu khung hình thời điểm t-1 có những thành phần giống khung hình trước thì chỉ những khác biệt mới được gửi đi. Khi một đối tượng di chuyển từ bức hình của khung hình đầu đến khung hình tiếp theo, nó sẽ xác định một vector chuyển động và vector này được gửi đến bộ mã hóa để sử dụng trong dự đoán vị trí. Với các đối tượng không chuyển động, ví dụ như đám mây trong hình, sẽ không cần gửi đi vector chuyển động, đây gọi là dự đoán liên khung (inter-prediction). Nếu dự đoán được hoàn thành trong 1 khung hình thì được gọi là dự đoán nội bộ (intra-prediction)

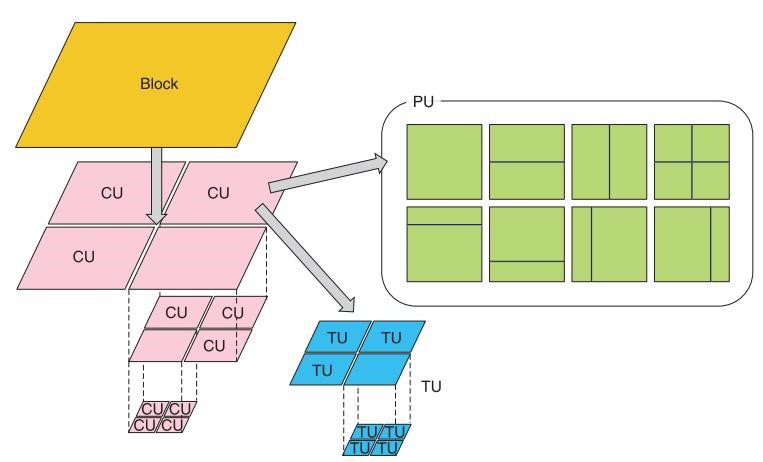


Hình 1.4 Dự đoán định hướng trong H.264 và H.265

​

Thay vì các macroblock 16x16 như trong H.264, H.265 phân vùng hình ảnh thành cây mã hóa (các CTB - ). Kích thước CTB có thể là 64x64, 32x32 hoặc 16x16, các nghiên cứu chỉ ra rằng CTB lớn giúp việc mã hóa có hiệu quả cao nhưng bên cạnh đó thời gian mã hóa cũng sẽ lớn. Bạn có thể tưởng tượng bạn có một file text 10GB, nếu chia file này thành 10 file kích thước 10GB, vậy khi nén 10 file này riêng lẻ thì hiệu quả nén sẽ không thể bằng nén riêng file 10GB nhưng cũng vì thế mà thời gian nén file 10GB sẽ lớn hơn tổng thời gian nén 10 file riêng lẻ.

Mỗi CTB có thể được phân chia một cách đệ quy trong một cấu trúc chia 4 bao gồm các khối nhỏ 32x32, 16x16 hoặc 8x8 gọi là đơn vị mã hóa vùng CU (xem hình minh họa dưới). CU là đơn vị cơ bản của các dự đoán trong HEVC, các đơn vị nhỏ được sử dụng trong các khu vực nhiều chi tiết như biên của một hình trong khi các đơn vị lớn được sử dụng trong các khu vực nội biên.



Hình 1.5 Phân chia các Block thành các TU

​

**c, Chuyển đổi và lượng tử**

Bất kỳ dữ liệu dư còn lại sau khi dự đoán được chuyển đổi bằng phép biến đổi cosine rời rạc (DCT) hoặc phép biển đổi sin rời rạc (DST). Một hoặc nhiều khối biến đổi kích thước 32×32, 16×16, 8×8 và 4×4 được áp dụng cho dữ liệu còn lại trong mỗi CU.  
**d, Mã hóa en-trô-pi (hệ số nhiễu loạn)**

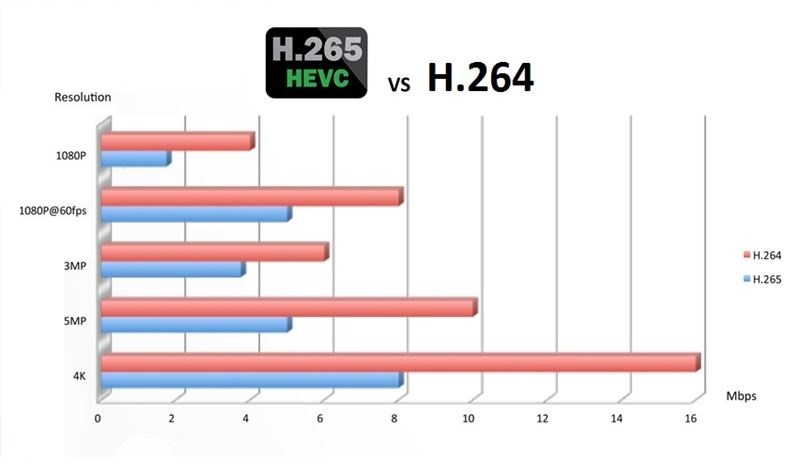
Một bitstream HEVC được mã hóa bao gồm chuyển đổi hệ số lượng tử, thông tin dự đoán như chế độ dự báo và vectơ chuyển động, thông tin phân vùng và dữ liệu khác… Tất cả những yếu tố này được mã hóa bằng cách sử dụng mà hóa nhị phân số học (CABAC).  
**e, Chất lượng H.265/HEVC so với H.264/AVC**

Hình bên dưới là một ví dụ so sánh H.265/HEVC và H.264/AVC đây là hai khung hình video ở độ phân giải HD 720p. Bên trái là sử dụng chuẩn nén H.264, bên phải sử dụng chuẩn nén H.265 với bitrate 420kbps. Có thể thấy chất lượng của video H.265 tốt hơn video H.264 khi các vùng màu thể hiện rõ ràng và sắc nét hơn, vùng tóc cũng thể hiện mượt mà hơn.



Hình 1.6 So sánh H.264/AVC và H.265/HEVC

Thời gian tải video H.265 so với video H.264 với cùng một tốc độ mạng. Chúng ta có thể thấy rõ được video chuẩn nén H.265 lợi như nào.



Hình . So sánh thời gian tải video giữa H265 và H264 khi cùng tốc độ mạng

​

## Chương trình truyền thông đa phương tiện VLC

### Giới thiệu tổng quát

**VLC media player** (thường gọi tắt là **VLC**) là một [media player](https://vi.m.wikipedia.org/wiki/Media_player) và [streaming media](https://vi.m.wikipedia.org/w/index.php?title=Streaming_media&action=edit&redlink=1) [server](https://vi.m.wikipedia.org/w/index.php?title=Server_(computing)&action=edit&redlink=1) [mã nguồn mở](https://vi.m.wikipedia.org/wiki/Ph%E1%BA%A7n_m%E1%BB%81m_t%E1%BB%B1_do_ngu%E1%BB%93n_m%E1%BB%9F) [di động cao](https://vi.m.wikipedia.org/wiki/Ph%E1%BA%A7n_m%E1%BB%81m_x%C3%A1ch_tay) và [đa nền tảng](https://vi.m.wikipedia.org/wiki/Cross-platform) được viết bởi [VideoLAN](https://vi.m.wikipedia.org/w/index.php?title=VideoLAN&action=edit&redlink=1) project.

LC hỗ trợ nhiều phương thức nén âm thanh và video cũng như nhiều định dạng file, bao gồm [DVD-Video](https://vi.wikipedia.org/wiki/DVD-Video), [Video CD](https://vi.wikipedia.org/wiki/Video_CD) và [giao thức](https://vi.wikipedia.org/wiki/Giao_th%E1%BB%A9c_truy%E1%BB%81n_th%C3%B4ng) streaming. Nó cho phép stream từ [mạng máy tính](https://vi.wikipedia.org/wiki/M%E1%BA%A1ng_m%C3%A1y_t%C3%ADnh) và [chuyển mã](https://vi.wikipedia.org/w/index.php?title=Chuy%E1%BB%83n_m%C3%A3&action=edit&redlink=1) các file multimedia.

Phân phối mặc định của VLC bao gồm một số lượng lớn các thư viện [decoding](https://vi.wikipedia.org/w/index.php?title=Decoding&action=edit&redlink=1) và [encoding](https://vi.wikipedia.org/w/index.php?title=Encoding&action=edit&redlink=1) miễn phí, tránh việc tìm kiếm/ bổ sung các plugin độc quyền. Nhiều trong số các [codecs](https://vi.wikipedia.org/w/index.php?title=Codecs&action=edit&redlink=1) của VLCđược cung cấp bởi thư viện [libavcodec](https://vi.wikipedia.org/w/index.php?title=Libavcodec&action=edit&redlink=1) từ dự án [FFmpeg](https://vi.wikipedia.org/w/index.php?title=FFmpeg&action=edit&redlink=1), nhưng nó dùng chủ yếu từ [muxer](https://vi.wikipedia.org/w/index.php?title=Muxer&action=edit&redlink=1) và demuxers riêng cũng như các giao thức riêng của nó. nó cũng được xem là media player đầu tiên hỗ trợ phát lại các đĩa DVD được mã hóa trên [Linux](https://vi.wikipedia.org/wiki/Linux) và [OS X](https://vi.wikipedia.org/wiki/OS_X) bằng cách sử dụng thư viện giải mã DVD [libdvdcss](https://vi.wikipedia.org/wiki/Libdvdcss).

### Tiến hành tải mã nguồn và cài đặt

Các bước build VLC từ source code trên Linux

Chương trình sau đây được thực hiện trên hệ điều hành Ubuntu

**Tải mã nguồn**

|  |
| --- |
| %wget <ftp://ftp.videolan.org/pub/videolan/vlc/2.2.2/vlc-2.2.2.tar.xz>  % tar xvJf vlc-2.2.2.tar.xz  % cd vlc-2.2.2 |

**Cài đặt các thư viện bên thứ 3**

|  |
| --- |
| $ sudo apt-get -y install libvorbis-dev libogg-dev libtheora-dev speex libspeex-dev flac libflac-dev x264 libx264-dev a52-0.7.4 liba52-0.7.4-dev mpeg2dec libmpeg2-4-dev faad libfaad-dev faac libfaac-dev lame libmp3lame-dev ffmpeg libavdevice-dev libmad0 libmad0-dev libschroedinger-dev libdca-dev twolame libtwolame-dev libmpcdec-dev libvorbisidec1 libvorbisidec-dev libass-dev libebml-dev libmatroska-dev libdvbpsi-dev libmodplug1 libmodplug-dev libshout3 libshout3-dev libdvdread4 libdvdnav4 libdvdnav-dev livemedia-utils liblivemedia-dev libcddb2 libcddb2-dev libcdio-dev libcdio-utils vcdimager libvcdinfo0 libvcdinfo-dev libgpg-error0 libgpg-error-dev libgcrypt11-dev gnutls-bin libgnutls-dev libdap-bin libdap-dev libxml2 libxml2-dev libpng12-0 libpng12-dev libjpeg8 libsdl1.2-dev libsdl-image1.2 libsdl-image1.2-dev libc-bin gettext libfreetype6 libfreetype6-dev libfribidi-dev libfribidi0 zlib1g zlib1g-dev libtag1-dev libcaca0 libcaca-dev caca-utils libqt4-dev libportaudio2 libupnp-dev libexpat1 libexpat1-dev yasm libxcb-xv0 libxcb-xv0-dev libx11-xcb1 libx11-xcb-dev libswscale-dev lua5.1 libxcb-composite0-dev  % sudo apt-get install libxcb-shm0-dev libxcb-xv0-dev libxcb-keysyms1-dev libxcb-randr0-dev libxcb-composite0-dev  % sudo apt-get install libx11-xcb-dev |

**Thực hiện cấu hình chương trình trước khi build**

|  |
| --- |
| % ./configure |

**Thực hiện compile chương trình**

|  |
| --- |
| % make |

Sau khi build thành công, chương trình VLC được đặt ngay ở thư mục build

**Ta khởi động chương trình**

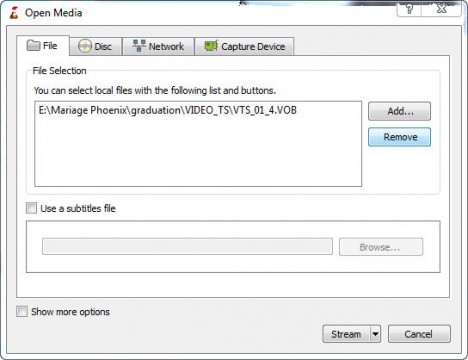
|  |
| --- |
| #./vlc |

# Thiết lập một hệ thống tạo luồng tín hiệu đa phương tiện cho tiếng và hình (video streaming) giữa hai máy tính (sử dụng chuẩn H265)

## Stream video trên một máy tính sử dụng VLC

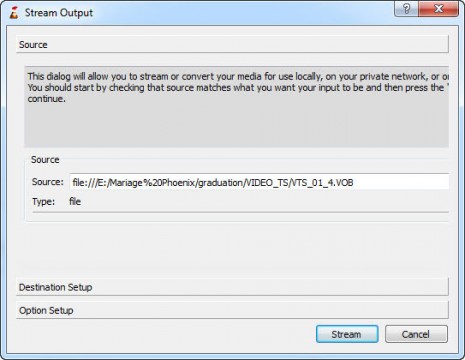
**1.**Bắt đầu mở VLC player. Điều hướng đến **Media** trên menu chính và từ trình đơn thả xuống, bấm vào **Stream…**

**2.**Một cửa sổ mới sẽ xuất hiện. Dưới tab **File**, click vào nút **Add** để thêm các tập tin bạn muốn stream. Bạn có thể chọn nhiều tập tin nếu bạn muốn VLC xây dựng  một danh sách nhạc. Khi bạn đang thực hiện thêm các tập tin, click vào **Stream.**

[](http://www.phanmem.com/wp-content/uploads/2013/05/video-streaming-vlc.jpg)

Hình . Khởi tạo server stream video

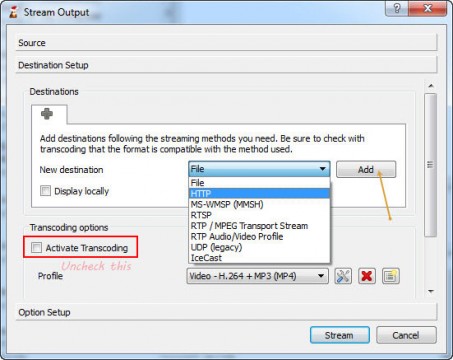
**3.**Cửa sổ***Stream Output***xuất hiện. Trong phần  ***Source.*** Hãy đảm bảo các tập tin là như nhau,mà bạn muốn stream. Bây giờ sau khi xác định điều này, kích ***Next.***

[](http://www.phanmem.com/wp-content/uploads/2013/05/video-streaming-vlc-player.jpg)

Hình . Chọn video muốn stream

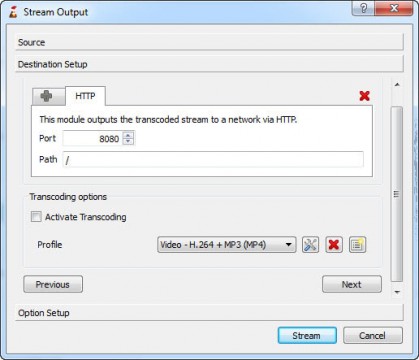
**4.** Dưới tab **Destinations**.  Bạn sẽ thấy một trình đơn thả xuống có chứa một danh sách các giao thức mà VLC hỗ trợ khi trình chiếu một video hoặc tập tin âm thanh. Chọn **HTTP** từ trình đơn.

Check tùy chọn **Display locally** nếu bạn muốn video hoặc tập tin âm thanh để chơi trên máy tính hiện tại là tốt nhất. Thông thường, bạn nên để nó là unchecked.

[](http://www.phanmem.com/wp-content/uploads/2013/05/streaming-protocols-vlc1.jpg)

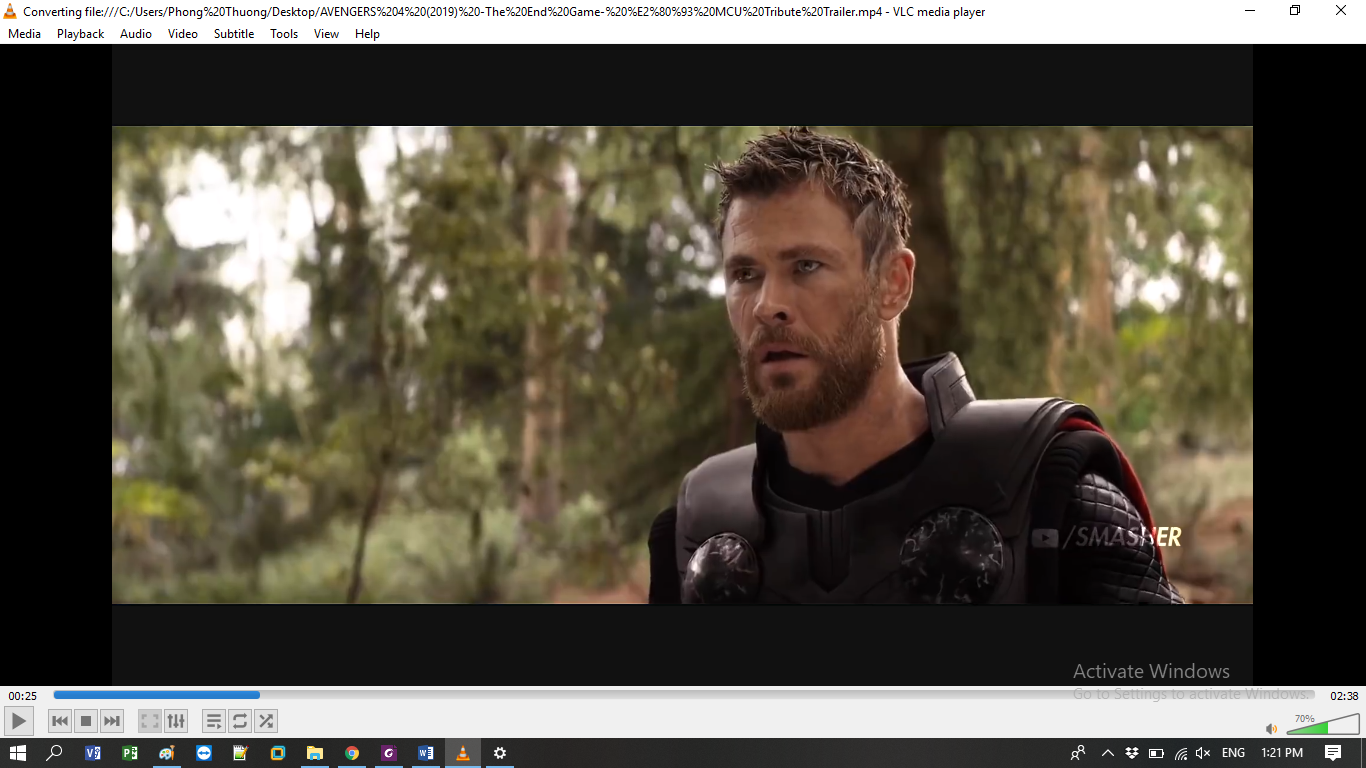
Hình . Chọn giao thức stream video

Bạn sẽ thấy một tùy chọn chuyển mã bên dưới được check theo mặc định. Bỏ chọn nó. Sau đó click **Add**. Bạn có thể rời khỏi phần thiết lập **Port và Path.**

[](http://www.phanmem.com/wp-content/uploads/2013/05/http-video-streaming.jpg)

Hình . Định dạng video muốn stream

Nhấn **Next**, để mở tab **Options**. Các cấu hình linh tinh là tùy chọn. Vì vậy, bạn có thể rời khỏi nó. Bây giờ hãy nhấp **Stream.**

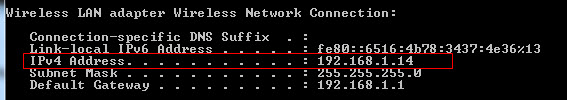


Hình . Video đang được stream

## Xem video streaming trên một máy tính khác

Bây giờ là thời gian để xem video stream trên các máy tính khác trong mạng của bạn, nhưng đầu tiên, bạn sẽ phải lấy địa chỉ IP của máy tính mà bạn đang streaming tập tin âm thanh hoặc video từ nó.

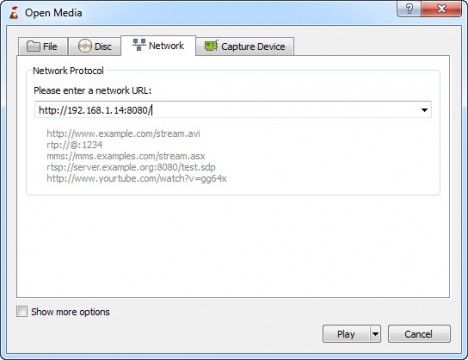
**5.** Nếu bạn không biết địa chỉ IP, bạn vào **Start> Run**, gõ **cmd** và nhấn **OK** để mở cửa sổ **Windows Command Prompt** của bạn. Sau đó, gõ lệnh **ipconfig** để có được local IP từ máy tính của bạn.

[](http://www.phanmem.com/wp-content/uploads/2013/05/get-local-ip-address-of-your-pc.jpg)

**6.**Một khi bạn có địa chỉ IP, viết lên một mảnh giấy và đi đến các máy tính khác (s) mà bạn muốn xem video stream. Mở VLC Player và điều hướng đến **Media> Open Network Stream**.

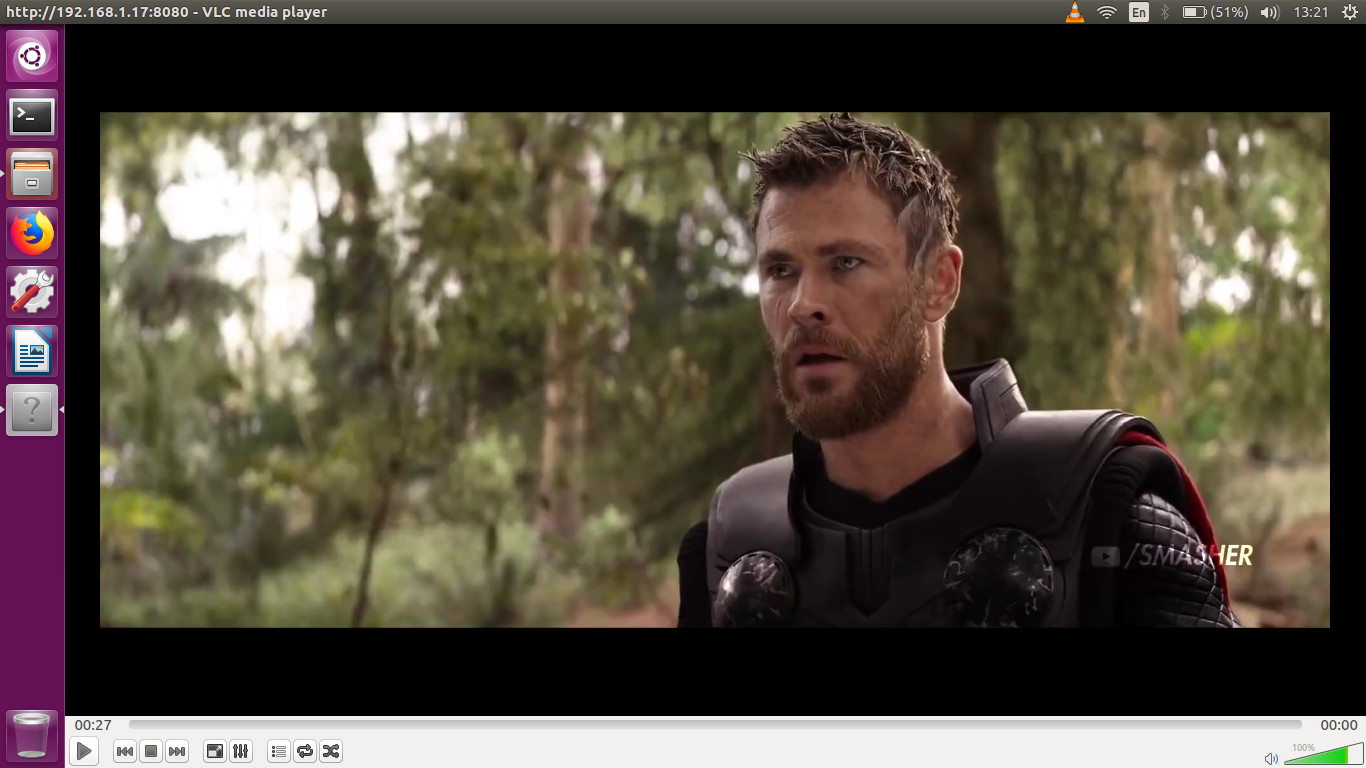
**7.**Nhập địa chỉ IP, số port và đường dẫn đến tập tin ảo theo các định dạng sau: **IP: PORT**

Trong hình trên, đó là **http://198.168.1.14:8080** và nhấn vào nút Play. Video hoặc âm thanh sẽ bắt đầu streaming sau một vài giây.

[](http://www.phanmem.com/wp-content/uploads/2013/05/watch-stream-videos-on-client-using-vlc-player.jpg)

Hình . Nhập đường dẫn đến tập tin ảo

Nếu bạn đã xác định một đường dẫn trong bước 4, sử dụng định dạng **IP: PORT / PATH.** Ví dụ, [**http://198.168.1.14:8080/**](http://198.168.1.14:8080/)



Hình . Video được hiển thị trên máy khác

* Đánh giá kết quả: Video bên máy thu sẽ bị delay một khoảng thời gian so với bên máy phát.

# Phát triển thêm tiện ích truyền văn bản nhanh giữa hai máy tính nói trên trong cùng giao diện GUI.

## Giới thiệu về Socket

### Khái niệm

“Socket là một cổng logic mà một chương trình sử dụng để kết nối với một chương trình khác chạy trên một máy tính khác trên Internet. Chương trình mạng có thể sử dụng nhiều Socket cùng một lúc, nhờ đó nhiều chương trình có thể sử dụng Internet cùng một lúc.”

Ở đây ta hiểu Socket trong Winsock như là một “phương tiện” để ứng dụng mạng có thể trao đổi dữ liệu. Nghĩa là 1 Server thì sẽ cần một Socket để lắng nghe, chờ đợi các kết nối từ client và Client thì phải cần có một Socket để kết nối tới Sever.

### Khởi tạo socket

**Hàm khởi tạo socket**

|  |
| --- |
| SOCKET socket (  **int** af,  **int** type,  **int** protocol  **);** |

**Ví dụ:**

|  |
| --- |
| SOCKET s = socket(AF\_INET,SOCK\_STREAM,IPPROTO\_IP)**;** |

Trong đó:

* af: Là một con số ID để quyết định Socket của chúng ta sử dụng giao thức (protocol) để kết nối.
* AF\_INET : TCP/IP (Phổ biến nhất hiện nay -> dùng địa chỉ IP để truyền dữ liệu)
* AF\_NETBIOS: NetBIOS (Giao thức dùng tên máy để truyền dữ liệu)
* AF\_APPLETALK: AppleTalk
* AF\_ATM: ATM

…

**Blind**

|  |
| --- |
| **int** bind(       SOCKET                     s,  **const** **struct** sockaddr FAR\* name,  **int**                        namelen  **);** |

Tác dụng dụng của BIND là sẽ giúp cho SOCKET của SERVER biết rằng nó sẽ chờ đợi kết nối và nhận dữ liệu trên IP nào và PORT nào.

Một số thông tin về các port:

* 0 -1023: Là những PORT đã được sử dụng bởi các dịch vụ như WEB, FTP, …
* 1024-49151: Là PORT dành cho SERVER lắng nghe. SERVER nên chọn trong khoảng này.
* 49152-65535: Là PORT khởi tạo ngẫu nhiên dành cho CLIENT kết nối tới Server.
* Hàm Bind gồm có 3 thông số:
* SOCKET s: Socket được thiết lập
* sockaddr name: Cấu trúc ADDR bao gồm địa chỉ IP và PORT
* int namelen: Kích thước của cấu trúc sockaddr

**Listen:**

|  |
| --- |
| **int** listen(      SOCKET s,  **int**    backlog  )**;** |

Kể từ khi gọi hàm này thì SERVER sẽ bắt đầu lắng nghe kết nối của mình.

Hàm LISTEN gồm có 2 thông số:

* SOCKET s: Socket đã được thiết lập IP và PORT.
* int backlog: Số kết nối cho phép chờ trong hàng đợi khi Server chưa chấp nhận kết nối. (vì đôi lúc có thể có tới 2 hay 3 client kết nối tới cùng 1 lúc). Giá trị tốt nhất là khoảng từ 5 – 10.

**Connect:**

|  |
| --- |
| **int** connect(         SOCKET s,  **struct** sockaddr \*serv\_addr,  **int** addrlen )**;** |

Hàm được gọi từ CLIENT nếu nó muốn kết nối tới SERVER:

* SOCKET s: Socket đã được khởi tạo.
* sockaddr \*serv\_addr: IP và PORT của Server.
* int addrlen: Sizeof của cấu trúc sockaddr.

**Accept**

|  |
| --- |
| SOCKET accept(       SOCKET  s,  **struct** sockaddr FAR\*  addr,  **int** FAR\*  addrlen  **);** |

Khi Client kết nối tới Server. Nó phải chờ Server chấp nhận kết nối (nếu ở giao thức TCP) bằng hàm accept trên.

Hàm ACCEPT gồm có 2 thông số:

* SOCKET s: Socket lắng nghe của SERVER.
* sockaddr addr: Là cấu trúc sockaddr lưu địa chỉ IP và PORT của CLIENT kết nối tới SERVER.
* int addrlen: Kích thước cấu trúc địa chỉ IP này.

Hàm ACCPET trả về 1 SOCKET mới Socket mới được tạo này đại diện cho 1 Connection (kết nối) mới giữa Server và Client. Sau khi đã truyền dữ liệu thì ta phải đóng SOCKET này lại bằng hàm close như closesocket (connect);

**Send/Receive**

|  |
| --- |
| **int** recv(      SOCKET s,  **char** FAR\* buf,  **int** len,  **int flags**  )**;** |

|  |
| --- |
| **int** send(      SOCKET s,  **const** **char** FAR \* buf,  **int** len,  **int flags**  )**;** |

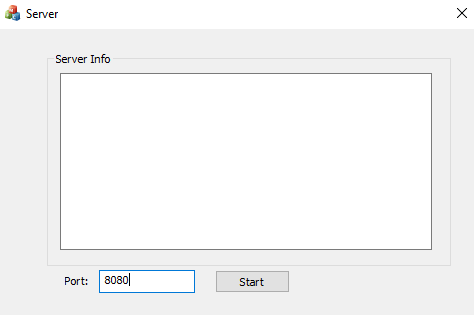
* SOCKET s: Là SOCKET được tạo ra khi Server chấp nhận kết nối từ CLIENT
* char FAR\* buf: Là dữ liệu (dạng BYTE – char) nhận hay gửi.
* int len: Kích thước của dữ liệu.
* int flags: Một số cờ hiệu đi kèm (thông thường là 0).

### Truyền tin nhắn văn bản nhanh giữa hai hay nhiều máy

Chúng em tạo chương trình Multi client chat room giữa server và client bằng ngôn ngữ C++ sử dụng MFC và WinSock.

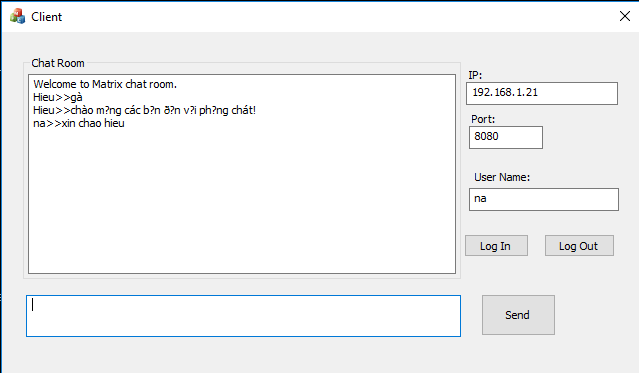
Server:

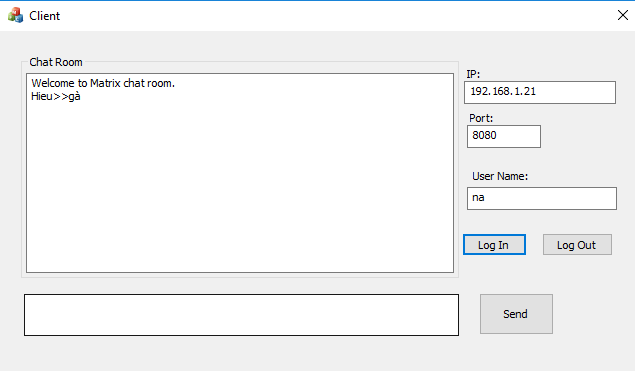
Trong máy chủ này, nhiều người dùng có thể đăng nhập vào và trò truyện với nhau như người dùng đa hướng. Trong thực tế nó có thể sử dụng máy chủ phòng chat. Đầu tiên ta cần tạo PORT để các client có thể truy cập vào, sau đó nhấn Start để khởi tạo phòng:



Client:

Các client cần đăng nhập địa chỉ IP và USER NAME của nó và nhập PORT của server để kết nối với server, sau đó nhấn join để vào phòng:





Như vậy chúng em đã thực hiện thành công truyền tin văn bản nhanh giữa hai máy trên cùng GUI.

# KẾT LUẬN

Qua đề tài “Streaming video với chuẩn video HEVC” này, chúng em đã hiểu rõ hơn chuẩn mã hóa H.265/HEVC về nguyên lý và cách thực hiện. Chúng em cũng đã biết thêm về phát video streaming thông qua sử dụng phần mềm VLC. Chúng em xin chân thành cảm ơn đến TS. Phạm Văn Tiến đã tận tình giúp đỡ bọn em hoàn thành đề tài này.

# TÀI LIỆU THAM KHẢO

[1] https://www.videolan.org,

truy nhập cuối cùng ngày 18/11/2018.

[2] <https://wiki.videolan.org>,

truy cập cuối cùng ngày 18/11/2018.

[3] <https://vi.m.wikipedia.org>,

Truy cập lần cuối ngày 18/11/2018.

[4] <https://tinhte.vn>,

Truy cập lần cuối ngày 18/11/2018.