

## ôMỤC LỤC

I.	GIỚI THIỆU ĐỀ TÀI	2
1.	Mô tả đề tài	2
2.	Tóm tắt nội dung báo cáo	3
II.	CƠ SỞ LÝ THUYẾT	3
1.	Streaming Video	3
a.	Streaming Video	3
b.	Quá trình Video Streaming	4
c.	Kiến trúc hệ thống Streaming	5
2.	Cache	5
3.	Các loại tập tin Video	7
4.	Định dạng tập tin FLV	9
a.	Tại sao chọn định dạng FLV?	9
b.	FLV	10
5.	Progressive download trong Flash	15
III.	TÌM HIỂU CÔNG NGHỆ	16
IV.	PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG	20
1.	Mô hình tổng quan	20
3.	Kiến trúc hệ thống	21
V.	THỰC HIỆN	27
1.	Server:	27
a.	Chờ kết nối :	27
b.	Đa luồng:	27
c.	Đọc tập tin:	28
2.	Client:	28
a.	Kết nối server:	28
b.	Tải video:	28
c.	Player:	28
3.	Giao diện:	29
a.	Server:	29
b.	Client:	29

VI.	TỔNG KẾT ĐỒ ÁN	30
VII.	PHƯƠNG HƯỚNG PHÁT TRIỂN LUẬN VĂN	31
1.	Nghiên cứu	31
a.	Quản lý Video Caching	31
b.	Quản lý Streaming	31
2.	Hiện thực	34
VIII.	TÀI LIỆU THAM KHẢO	34
NHẬT KÍ LÀM VIỆC ĐỒ ÁN MÔN HỌC 2		35

## I. GIỚI THIỆU ĐỀ TÀI

### 1. Mô tả đề tài

Ngày nay, việc trình chiếu Video trên các website trở nên rất phổ biến, bên cạnh đó việc sử dụng bài giảng điện tử cho chương trình đào tạo từ xa cũng rất phù hợp, giúp nâng cao chất lượng và hiệu quả đào tạo. Tuy nhiên, việc trình chiếu Video trực tuyến hiện nay vẫn còn khá nhiều hạn chế: mặc dù ngày nay băng thông mạng đã được mở rộng nhưng đường truyền vẫn không thể đáp ứng được nhu cầu sử dụng ngày càng tăng về lưu lượng truy cập, cũng như chất lượng Video; việc quản lý và bảo vệ bản quyền gặp khó khăn vì người dùng dễ dàng sử dụng các chương trình tự động kết nối và tải Video về.

Việc nghiên cứu tìm phương pháp giải quyết các vấn đề trên đã được rất nhiều sự quan tâm. Ý tưởng đưa ra là xây dựng hệ thống Media caching (đặc biệt là Video caching) trên server để thay thế cho

web caching đang quá tải hiện nay. Các nghiên cứu cho thấy kỹ thuật Video caching có nét tương đồng với web caching, thêm vào đó, là những qui định ràng buộc về truyền nhận dữ liệu được giải quyết hoàn toàn bên phía server, thông qua các số liệu thống kê lượt gửi nhận yêu cầu từ phía Client.

Trên thế giới có nhiều dự án nghiên cứu để giải quyết cho các website trình chiếu Video phổ biến hiện nay. Sự ra đời của những sản phẩm thương mại đầu tiên hỗ trợ Media caching (như phần mềm của UltraBand PeerApp) đã đem lại những cải thiện rất lớn và chứng minh tính khả thi của hướng tiếp cận trên.

Nghiên cứu các phương thức lưu trữ và kỹ thuật quản lý dữ liệu trên server và cache server. Sử dụng kỹ thuật Video caching, giúp hỗ trợ tốt việc trình chiếu dữ liệu, cải thiện hiệu suất người dùng cuối; tiết kiệm được băng thông và việc đầu tư mạng lưới cơ sở hạ tầng; và hạn chế việc vi phạm bản quyền dữ liệu (Video, Slide) tuân thủ đúng luật áp dụng bảo vệ trách nhiệm cung cấp của các tổ chức uy tín.

Nghiên cứu và xây dựng một hệ thống hỗ trợ trình chiếu Video sử dụng kỹ thuật Video caching tích hợp vào website đào tạo trực tuyến là một nhu cầu cần thiết và đem lại rất nhiều hiệu quả.

Tầm vực của đề tài là tạo một server có hỗ trợ Video caching để tăng tốc độ truyền tải Video đến người dùng mà không hạn chế về dung lượng hay đường truyền, với những mạng dial-up cũng xem được Video với chất lượng tốt mà không bị gián đoạn.

Để giải quyết bài toán trên, nhóm thực hiện chia làm hai giai đoạn: Đồ án 2 và luận văn. Trong giai đoạn đồ án 2 chủ yếu là tìm hiểu những kiến thức liên quan, nghiên cứu giải pháp giải quyết vấn đề, thử nghiệm những công cụ phát triển hệ thống; tập trung xử lý quản lý dữ liệu phía Client và truyền tải dữ liệu từ Server về Client.

## ***2. Tóm tắt nội dung báo cáo***

Tài liệu này là kết quả những vấn đề tìm hiểu trong giai đoạn đồ án 2 gồm:

- Chương II: Tìm hiểu những kiến thức cơ sở: Streaming Video, Progressive download, Video Caching.
- Chương III: Tìm hiểu và thử nghiệm những công nghệ để hiện thực: ActionScript cho Client, Java cho Server.
- Chương IV: Phân tích và thiết kế hệ thống, tài liệu mô tả về nghiệp vụ, đưa ra kiến trúc hệ thống.
- Chương V: Thực hiện.
- Chương VI: Tổng kết những công việc đã làm trong giai đoạn đồ án.
- Chương VII: Đề xuất phương hướng nghiên cứu và phát triển luận văn.

## II. CƠ SỞ LÝ THUYẾT

### 1. *Streaming Video*

#### a. *Streaming Video*

- Streaming Video là một phương thức để máy chủ đặc biệt cung cấp Video thông qua Internet. Streaming Video thường được sử dụng trong lĩnh vực giải trí hoặc dạy học, dùng để lưu trữ các tuyến tập các tập tin Video hoặc các bài học, cung cấp cho người dùng các tiện ích như tìm kiếm, liệt kê, và khả năng hiển thị hoặc hiển thị lại các dữ liệu Video theo yêu cầu.
- Với các định dạng tập tin Video truyền thống, dữ liệu chỉ có thể hiển thị khi đã được tải về (download) toàn bộ, vì vậy đối với các tập tin Video chất lượng cao có dung lượng lớn thì công việc này sẽ tiêu tốn rất nhiều thời gian.
- Streaming Video tiết kiệm thời gian cho người dùng bằng cách sử dụng các công nghệ giải nén kết hợp với player hiển thị dữ liệu đồng thời trong lúc vẫn tiếp tục download. Quá trình này được gọi là buffering:
  - + Thay vì được gửi một lần duy nhất, dữ liệu streaming Video sẽ được truyền đi thành các gói nhỏ.
  - + Ban đầu player sẽ lấy về một phần chia nhỏ đó của dữ liệu Video trước khi hiển thị, đồng thời trong lúc hiển thị các gói dữ liệu còn lại sẽ lần lượt được lấy về để kịp cho việc hiển thị tiếp theo.
- Streaming Video được thể hiện dưới hai dạng:
  - + Video theo yêu cầu (on demand): là các dữ liệu Video được lưu trữ trên multimedia server và được truyền đến người dùng khi có yêu cầu, người dùng có toàn quyền để hiển thị cũng như thực hiện các thao tác (tua, dừng,qua ...) với các đoạn dữ liệu này.
  - + Video thời gian thực (live event): là các dữ liệu Video được convert trực tiếp từ các nguồn cung cấp dữ liệu theo thời gian thực (máy camera, microphone, thiết bị phát dữ liệu Video...).

**b. Quá trình Video Streaming**

- Video Streaming là thuật ngữ dùng để nói về quá trình máy chủ cung cấp Video thông qua Internet. Với Streaming, máy tính của người sử dụng không phải tải toàn bộ Video về cùng một lúc. Để thực hiện việc tăng tốc độ lần tải về, tại bất kỳ thời điểm nào máy tính chỉ cần tải về một phần thông tin của Video. Với kỹ thuật này, chúng ta cần một máy chủ đặc biệt kiểm soát việc cung cấp các nội dung Video. Để bất kỳ phần nào của đoạn Video được lưu trữ trên server có thể được truy cập bất kỳ lúc nào, hơn là cần nó chờ để tải về trước khi truy cập nó.
- Video Streaming:
  - + True Streaming: tín hiệu Video đến theo thời gian thực và hiện thị ngay lập tức cho người xem.
  - + Download and Play: Sau khi tải toàn bộ Video về, sau đó phát Video.
  - + Progressive Download and Play: Đây là công nghệ lai giữa hai công nghệ trên. Trong công nghệ này, Video sẽ được chia nhỏ ra thành nhiều phần nhỏ; chương trình phát Video (player) sẽ hiện thị ngay những segment vừa được tải về và lưu trữ dữ liệu đó ở ổ đĩa cục bộ của máy. Khi tải lại những đoạn Video mà đã tải về, Player sẽ lấy dữ liệu từ ổ đĩa cục bộ để tải lên.
- Video Streaming sử dụng các giao thức RTP, MMS, HTTP.. để truyền dữ liệu theo dạng streaming qua mạng Internet, đồng thời sử dụng các chuẩn nén để giảm dung lượng dữ liệu, cung cấp khả năng nén dữ liệu tại nhiều mức nén, nhiều kích thước hiển thị để có thể phù hợp với độ rộng băng thông của nhiều mạng truyền dẫn để tối ưu hoá việc truyền dữ liệu qua mạng. Cũng chính vì vậy việc truyền các Streaming Video qua mạng sẽ phụ thuộc rất nhiều vào các sản phẩm phần mềm Video Streaming Server. Hiện nay, có rất nhiều chuẩn công nghệ Video streaming được phát triển bởi các player khác nhau. Cũng như có nhiều định dạng dữ liệu riêng với các chuẩn công nghệ đó. Các định dạng Video streaming chỉ giới hạn bởi các công ty dẫn đầu trong công nghệ streaming: Các hãng này đều cung cấp các bộ công cụ trọn gói gồm Video Streaming Server (lưu trữ, truyền phát dữ liệu theo các giao thức hỗ trợ ...), Video Player (hiển thị dữ liệu tại phía người dùng), và công cụ kiến tạo dữ liệu với các chuẩn nén.

*c. Kiến trúc hệ thống Streaming*

- Khi hiện thực một hệ thống Streaming trên một mạng IP( hoặc một vài kỹ thuật khác), chúng ta cần đầu tư cho hệ thống một Streaming Server (có nhiệm vụ phân phối Stream cho mỗi thiết bị người dùng).
- Streaming Server hoạt động sẽ lấy nội dung (những tập tin Video hoặc những tập tin khác trong hệ thống) và tạo Stream cho mỗi yêu cầu người dùng gửi đến. Những Stream này có thể hiện thực theo hai phương thức (mỗi phương thức được điều khiển bởi cơ chế khác nhau) là unicast và multicast. Phổ biến nhất hiện nay là cơ chế unicast.
- Chức năng thường được tích hợp trong Streaming Server là lưu trữ và phục hồi nội dung. Công việc xử lý quan trọng của Streaming Server là tạo một gói tin cho mỗi Stream theo thời gian thực. Ngoài ra, Streaming Server còn phải giải quyết các yêu cầu về mã hóa gói dữ liệu để đảm bảo tính bảo mật, nén dữ liệu để gửi đến những kênh Stream có tùy chọn kết nối với tốc độ khác nhau. Một vấn đề quan trọng khác trong hệ thống mà nhóm nghiên cứu là phân tích dữ liệu và đưa ra giải thuật giải quyết vấn đề quản lý thiết bị Cache một cách hiệu quả trong Streaming Server.

**2. Cache**

- Với những hệ thống máy chủ trình chiếu Video trực tuyến hoặc cung cấp cho việc tải về theo yêu cầu người dùng đang được sử dụng hiện nay, thì với mỗi yêu cầu của người dùng đến - sẽ được máy chủ xử lý thông tin đảm bảo yêu cầu đó là hợp lệ để thiết lập kết nối; tiếp sau đó, nó sẽ truy xuất đến ổ cứng (hoặc thiết bị lưu trữ cục bộ) để truy xuất tập tin yêu cầu đó và gửi về cho người dùng. Như chúng ta đều biết - tốc độ truy xuất trên ổ cứng thường là khá chậm. Bên cạnh đó, có những tập tin được nhiều người dùng cùng quan tâm trong một thời điểm (sai khác nhỏ về thời gian), việc truy xuất ổ cứng liên tục để truy xuất cùng một tập tin như cách thức làm việc hiện tại là rất lãng phí và tốn kém cho máy chủ.
- Để giải quyết vấn đề này, ta đưa ra giải pháp sử dụng Cache. Cache là một cơ chế lưu trữ tốc độ cao đặc biệt. Cache – được hiểu theo nghĩa thường là nơi lưu trữ dữ liệu nằm chờ ứng dụng hoặc phần cứng xử lý, nhằm mục đích tăng tốc độ xử lý. Cache có thể là một vùng lưu trữ của bộ nhớ chính hoặc một thiết bị lưu trữ tốc độ cao độc lập. Có rất nhiều loại Cache được biết đến hiện nay với những chức năng khác nhau như: Cache của CPU, Caching của Internet Browser, Caching của Oracle... Cụ thể ở đây ta quan tâm đến một cơ chế xử lý Video

Caching ngay trên Server thay thế cho Web Caching quá tải hiện nay. Việc đầu tư cho thiết bị Cache cộng với việc có một giải pháp quản lý Cache hiệu quả sẽ giảm tải rất nhiều cho máy chủ.

— Cách thức hoạt động thông qua Cache:

- + Dữ liệu được truy xuất từ ổ cứng được đẩy lên Cache.
- + Với mỗi yêu cầu từ người nhận gửi tới, máy chủ sẽ tìm kiếm dữ liệu đã tồn tại trên Cache chưa để thực hiện tác vụ đẩy dữ liệu đó lên Cache; nếu đã tồn tại thì dữ liệu từ Cache - thông qua quá trình xử lý dữ liệu sẽ được truyền trực đến người dùng thông qua một Stream (bỏ qua giai đoạn truy xuất từ ổ cứng).
- + Cơ chế để quản lý dữ liệu trên Cache, quyết định dữ liệu nào được đưa vào Cache, hoặc loại bỏ ra khỏi Cache, chúng ta phải thông qua việc phân tích, khai phá thông tin truy xuất dữ liệu, cùng với những thử nghiệm những số liệu thực tế để đưa ra một giải thuật quản lý Cache hợp lý – công việc này của đề tài sẽ được nghiên cứu kỹ hơn trong giai đoạn luận văn.

### 3. Các loại tập tin Video

- Các tập tin Video thường thuộc 2 loại: codec, không codec. Một số định dạng tập tin, chỉ cần nhìn vào đuôi tập tin ta biết được loại codec mà tập tin đó sử dụng như: WMV, RM, MOV... Hoặc các loại đuôi tập tin không chỉ định chính xác loại codec tập tin nào như: MP4, AVI, MKV... Các loại tập tin trên đều là các container để chứa nội dung, chúng ta thường phải sử dụng các trình media player để mở và đọc các tập tin đó để biết tập tin đó sử dụng loại codec nào để mã hóa.
- Xem xét một số loại định dạng tập tin mà các website và các nhà cung cấp lớn thường hỗ trợ:

Định dạng	Giới thiệu
3GP – 3GPP Multimedia	+ Định dạng 3GP là định dạng Video được phát triển bởi dự án 3rd Generation Partnership dựa trên chuẩn MPEG-4. Được sử dụng cho các tập tin đa phương tiện trên mạng không dây 3G tốc độ cao, sử dụng phổ biến trên điện thoại có hỗ trợ quay phim.
ASF – Advanced Systems Format	+ Định dạng ASF được Microsoft phát triển. Định dạng này được dùng để truyền tải các tập tin đa phương tiện chứa văn bản, đồ họa, âm thanh, Video và hoạt họa. Tập tin ASF chủ yếu là tập tin Windows Media Audio và Windows Media Video.

	<ul style="list-style-type: none"> <li>+ Tập tin ASF chỉ đặc tả cấu trúc của tập tin audio hay âm thanh được truyền tải mà không chỉ rõ phương pháp mã hóa.</li> </ul>
AVI – Audio Video Interleave	<ul style="list-style-type: none"> <li>+ Định dạng AVI cũng được phát triển bởi Microsoft và chứa dữ liệu được mã hóa theo các codec khác nhau( có thể dùng các codec như DivX để mã hóa tập tin AVI). Định dạng AVI bao gồm hai loại định dạng chính DivX, Xvid. Đây là một định dạng rất phổ biến hiện nay.</li> <li>+ AVI là một sản phẩm của RIFF. Dữ liệu của tập tin được chia trong các khối, mỗi khối được xác định bởi một nhãn. Khối đầu tiên chứa dữ liệu về Video, những thông tin về chiều dài, chiều rộng, tỷ lệ khung. Khối thứ 2 chứa dữ liệu thật sự của tập tin AVI. Khối thứ 3 chứa các chỉ mục của các khối dữ liệu trong tập tin.</li> </ul>
FLV – Flash Video	<ul style="list-style-type: none"> <li>+ Flash Video là một định dạng Video thường được dùng cho những website chia sẻ như Youtube hoặc những website chiếu phim trực tuyến, được phát triển bởi Adobe System( Macromedia). Hỗ trợ 2 loại định dạng FLV và F4V.</li> <li>+ Flash Video: nội dung nhúng trong tập tin SWF và phần mở rộng FLV chứa thông tin tập tin.</li> <li>+ Định dạng này có chất lượng chấp nhận được, trong khi kích thước tập tin nhỏ, nên rất hiệu quả trong môi trường Internet. Phù hợp với hệ thống trình chiếu Video trực tuyến nói chung và hệ thống dạy học bằng Video nói riêng.</li> </ul>
MP4 – MPEG-4 Video	<ul style="list-style-type: none"> <li>+ Định dạng MPEG-4 là một tiêu chuẩn được phát triển bởi Moving Picture Experts Group, được sử dụng trên rất nhiều điện thoại và các thiết bị chơi Video (còn gọi là thiết bị chơi MP4).</li> </ul>
MOV – Apple QuickTime Movie	<ul style="list-style-type: none"> <li>+ QuickTime Movie là định dạng được Apple phát triển. Cũng là một định dạng thường được sử dụng trên Internet.</li> </ul>
RM – Real Media	<ul style="list-style-type: none"> <li>+ Real Media là định dạng được phát triển bởi RealNetworks. Real Media chứa cả thông tin về Video (Real Video) và âm thanh (Real Audio). Thường được dùng để truyền tải các tập tin đa phương tiện thông qua internet.</li> </ul>
VOB – DVD Video Object	<ul style="list-style-type: none"> <li>+ Định dạng VOB liên quan đến DVD Video Movie File. Một tập tin VOB thường chứa các luồng đa công (multiplex) gồm: Video, âm thanh và phụ đề.</li> </ul>



WMV- Windows Media Video	<ul style="list-style-type: none"><li>+ Windows Media Video là một định dạng tập tin thường gặp nhất cũng được phát triển bởi Microsoft. Tập tin Windows Media chứa Video được mã hóa theo bộ codec Windows Media Video và âm thanh được mã hóa theo codec Windows Media Audio codec.</li><li>+ Tập tin WMV được thiết kế dùng cho ứng dụng luồng (thread) trên Internet. Dung lượng tập tin WMV khá thấp nên dễ dàng truyền tải trên mạng, tuy nhiên chất lượng hình ảnh và âm thanh không thực sự tốt.</li></ul>
-----------------------------------	--

#### 4. Định dạng tập tin FLV

##### a. Tại sao chọn định dạng FLV?

— Lựa chọn định dạng tập tin FLV để hỗ trợ chính trong hệ thống vì :

- + Định dạng FLV có khả năng nén tốt nên kích thước một tập tin định dạng FLV là nhỏ nhất trong các loại tập tin cùng truyền tải một lượng nội dung( mà vẫn có chất lượng tập tin tốt) nên rất phù hợp với trình chiếu Video trực tuyến trên website.
- + Một trong những lợi ích chính của FLV là Progressive download, chúng ta có thể bắt đầu xem một đoạn Video mà không phải tải toàn bộ về trước, đó là lý do vì sao những website chia sẻ Video như Youtube đã rất thành công với định dạng này.
- + Hiệu suất và tốc độ tải dữ liệu trên web của FLV nhanh.
- + Định dạng FLV rất thuận lợi để sử dụng Streaming Video theo thời gian thực trên web.
- + Ngoài ra, hiện nay các trình duyệt phổ biến (IE, FireFox, Opera, Gchrome...) đều hỗ trợ flash player plug-in, FLV còn được hỗ trợ trên những trình chơi Video trên các hệ điều hành.
- + Cuối cùng, định dạng Flash Video là loại định dạng có tính bảo mật cao vì không thể chỉnh sửa hoặc chuyển mã (transcoded) như định dạng khác. Nội dung của FLV phải được decode từ SWF. Muốn thay đổi nội dung của FLV phải thay đổi từ SWF (đảm bảo toàn vẹn dữ liệu).
- + Sự ra đời của phiên bản HTML mới là HTML5 đã giới thiệu một thành phần mới là HTML5 Video (tag Video), người ta dần chuyển từ FLV sang MPEG-4 hoặc Windows Media cho các định dạng Video trên Web. Tuy nhiên, hiện tại FLV vẫn là định dạng Video tốt nhất cho việc trình chiếu Video trực tuyến. Với sự phát triển mạnh của HTML5 trong tương lai, Adobe đang phát triển những công cụ hỗ trợ chuyển đổi định dạng tập tin để thích hợp trên nền HTML5.

- Để thuận lợi cho người sử dụng là người cung cấp tập tin (tải tập tin lên website), hệ thống xây dựng cần tích hợp thêm phần mềm hỗ trợ chuyển đổi các định dạng được dùng phổ biến hiện nay như: AVI, MOV, WMV, RM... sang định dạng FLV.

*b. FLV*

- Chức năng(Function): Một tập tin xây dựng theo cấu trúc tập tin FLV có nghĩa là tập tin đó là một Video được làm bằng công nghệ Adobe Flash.
- Source: Chúng ta có thể sử dụng các chương trình như Asobe Flash CS5 (plug-in Flash Video) (version  $\geq 6$ ), Apple QuickTime Pro Video hoặc những phần mềm có tương thích với QuickTime Pro để tạo tập tin cấu trúc FLV.
- Reader Programs:
  - + Video LAN VLC media player
  - + Eltima Elmedia Player...
- Kiến trúc định dạng FLV(Format):
  - + Tiêu đề(header): cung cấp thông tin về tập tin(như: tên, ngày tháng).
  - + Gói Metadata: chứa dữ liệu AMF(những hướng dẫn để trình đọc Video phát tập tin như: kích thước, tốc độ yêu cầu).
  - + Nội dung (gói audio và Video được lưu trữ giống như trong SWF):
    - o Âm thanh(audio)
    - o Hình ảnh chuyển động(moving picture) (Video).
- HTTP Streaming:
  - + Semi-stream FLV thông qua http bằng thủ thuật gửi các tiêu đề(normal header) Di chuyển đến vị trí mới dựa vào timestamp.
  - + Hiện tại có một công cụ để streaming tập tin FLV và FLV4PHP.
- Overview:
  - + Chuẩn codec được sử dụng tương thích với hầu hết các tập tin FLV và H.264.
  - + Các tập tin theo định dạng FLV thì các tag (được lưu theo dạng index) hỗ trợ seek point (chuyển đến 1 điểm bất kỳ) (thông tin này được lưu trữ trong metadata).

- Thành phần:
  - + Metadata:

Property	Value	Notes
----------	-------	-------

duration	Obvious.	Unlike for FLV files this field will always be present.
Videocodecid	For H.264 it reports 'avc1'.	
audiocodecid	For AAC it reports 'mp4a', for MP3 it reports '.mp3'.	
avcprofile	66, 77, 88, 100, 110, 122 or 144	Corresponds to the H.264 profiles
avclevel	A number between 10 and 51.	Consult this list to find out more.
aotype	Either 0, 1 or 2.	This corresponds to AAC Main, AAC LC and SBR audio types.
moovposition	int	The offset in bytes of the moov atom in a file.
trackinfo	Array	An array of objects containing various information about all the tracks in a file.
chapters	Array	Information about chapters in audiobooks.
seekpoints	Array	Times you can directly feed into NetStream.seek();
Videoframerate	int	The frame rate of the Video if a monotone frame rate is used. Most Videos will have a monotone frame rate.
audiosamplerate		The original sampling rate of the audio track.
audiochannels		The original number of channels of the audio track.
tags		ID3 like tag information

+ FLV Header

Field	Data Type	Example	Description
Signature	byte[3]	"FLV"	Always "FLV"

Version	uint8	“\x01” (1)	Currently 1 for known FLV files
Flags	uint8 bitmask	“\x05” (5, audio+Video)	Bitmask: 4 is audio, 1 is Video
Offset	uint32_be	“\x00\x00\x00\x09” (9)	Total size of header (always 9 for known FLV files)

+ FLV Stream

Field	Data Type	Example	Description
PreviousTagSize	uint32_be	“\x00\x00\x00\x00” (0)	Always 0

Ta có một chuỗi các tag sau(theo kích thước tập tin)

+ FLV Tag

Field	Data Type	Example	Description
Type	uint8	“\x12”(0x12, META)	Determines the layout of Body, see below for tag types
BodyLength	uint24_be	“\x00\x00\xe0” (224)	Size of Body (total tag size - 11)
Timestamp	uint24_be	“\x00\x00\x00” (0)	Timestamp of tag (in milliseconds)
TimestampExtended	uint8	“\x00” (0)	Timestamp extension to form a uint32_be. This field has the upper 8 bits.
StreamId	uint24_be	“\x00\x00\x00”(0)	Always 0
Body	byte[BodyLength]	...	Dependent on the value of Type

+ Previous tag size

Field	Data Type	Example	Description
PreviousTagSize	uint32_be	“\x00\x00\x00\x00” (0)	Total size of previous tag, or 0 for first tag

+ FLV Tag Type

Tag code	Name	Description
0x08	AUDIO	Contains an audio packet similar to a SWF SoundStreamBlock plus codec information
0x09	VIDEO	Contains a Video packet similar to a SWF VideoFrame plus codec information
0x18	META	Contains two AMF packets, the name of the event and the data to go with it

- + FLV Tag 0x08: AUDIO Những byte đầu tiên của gói audio chứa biến cờ (bitflags) để mô tả codec được sử dụng:

Name	Expression	Description
soundType	(byte & 0x01) » 0	0: mono, 1: stereo
soundSize	(byte & 0x02) » 1	0: 8-bit, 1: 16-bit
soundRate	(byte & 0x0C) » 2	0: 5.5 kHz, 1: 11 kHz, 2: 22 kHz, 3: 44 kHz
soundFormat	(byte & 0xf0) » 4	0: Uncompressed, 1: ADPCM, 2: MP3, 5: Nellymoser 8kHz mono, 6: Nellymoser, 11: Speex

- + Phần còn lại của gói tin audio là dữ liệu theo 1 soundstreamblock SWF

- + FLV Tag 0x09: VIDEO. Những byte đầu tiên của gói Video chứa biến cờ (bitflags) để mô tả codec được sử dụng và các loại frame:

Name	Expression	Description
------	------------	-------------

codecID	(byte & 0x0f) » 0	2: Sorensen H.263, 3: Screen Video, 4: On2 VP6, 5: On2 VP6 Alpha, 6: ScreenVideo 2
frameType	(byte & 0xf0) » 4	1: keyframe, 2: inter frame, 3: disposable inter frame

Thông tin về frameType để hỗ trợ decode nội dung của gói Video trong trường hợp metadata có lỗi.

- + FLV Tag 0x18: META Nội dung của một gói META là 2 gói AMF.
  - 1 gói là dạng thu gọn: uint16\_be length-prefixed UTF-8 string (AMF type 0x02)
  - Gói còn lại: chứa dạng mảng hỗn hợp (AMF type 0x08) – có nghĩa là chứa 1 loạt các AMF khác.

### 5. *Progressive download trong Flash*

- Nếu bạn không thể truy cập đến Flash Communication Server hoặc FVSS, bạn vẫn có thể tận hưởng những lợi ích của việc loading Video từ một nguồn bên ngoài bằng cách sử dụng Progressive download. Progressive download là kỹ thuật NDS duy nhất cho phép người sử dụng đạt hiệu quả chi phí tải về những tập tin theo yêu cầu có kích thước lớn trên những đường truyền Internet có băng thông thấp mà không cần máy chủ Video tồn kém.
- Progressive download một Video từ một web server thì cung cấp *hiệu suất thời gian thực* giống như Flash Communication Server; ngoài ra, bạn có thể dùng các Video có kích thước tương đối lớn, và giữ cho kích thước tập tin SWF được publish của bạn đến mức tối thiểu. Bạn cũng có thể dùng thành phần FLV Playback mới hoặc ActionScript để điều khiển việc thu phát Video và cung cấp điều khiển trực quan cho người sử dụng tương tác với Video.
- Cách thức:
  - + Progressive download cho phép sử dụng ActionScript để tải một tập tin dạng FLV từ bên ngoài vào một tập tin SWF. Sau đó phát tập tin đó theo thời gian thực.
  - + Cụ thể là sử dụng netConnection và các đối tượng netStream để phát lại các tập tin FLV và kiểm soát được các chức năng như Play, Pause, Seek một Video cũng như là thời gian và kích thước phần bộ đệm dành cho tập tin Video đó.
  - + Một điểm lưu ý ở đây, nội dung của tập tin Video đó được lưu trữ bên ngoài nội dung của Flash và cả chương trình phát Video (player) đó.
- Ưu điểm:
  - + Bạn chỉ cần đưa ra giao diện SWF để người dùng xem trước và kiểm tra để từ đó chọn xem phần nội dung Flash.

- + Video được chia ra thành nhiều segment, và chương trình phát Video (player) sẽ chơi ngay những segment vừa được tải về và lưu trữ dữ liệu đó ở ổ đĩa cục bộ của máy.
  - + Khi chạy, chương trình phát sẽ tải những tập tin từ trong ổ đĩa của máy tính vào các tập tin SWF với không gian giới hạn kích thước của tập tin và thời gian.
  - + Khung nhìn (Interface) của tập tin Video có thể khác biệt với tập tin SWF.
- Nhược điểm:
- + Nội dung phải được phân đoạn theo định dạng của Progressive Download, mà hiện tại chỉ có một số ít Media Player hỗ trợ nó.
  - + Phải tốn nhiều chi phí và phức tạp trong việc quản lý nhiều tập tin trên Server.
  - + Progressive Download chỉ làm việc tốt khi tốc độ đường truyền Internet phải lớn hơn hoặc bằng tốc độ của Stream.

### III. TÌM HIỂU CÔNG NGHỆ

— Java:

- + Java là một công nghệ xây dựng các ứng dụng phần mềm được coi là công nghệ mang tính cách mạng và khả thi nhất trong việc tạo ra các ứng dụng có khả năng chạy thống nhất trên nhiều nền tảng mà chỉ cần biên dịch một lần. Java vừa là ngôn ngữ lập trình được giới thiệu bởi Sun Microsystems, còn là một nền tảng phát triển và triển khai ứng dụng trong đó máy ảo Java, bộ thông dịch có vai trò trung tâm. Java được xây dựng trên nền tảng của C và C++. Do vậy nó sử dụng các cú pháp của C và các đặc trưng hướng đối tượng của C++. Java là ngôn ngữ lập trình hướng đối tượng. Java là ngôn ngữ vừa biên dịch vừa thông dịch. Đầu tiên mã nguồn được biên dịch bằng công cụ JAVAC để chuyển thành dạng ByteCode(file class). Sau đó được thực thi trên từng loại máy cụ thể nhờ chương trình thông dịch. Có nghĩa là khi Compile một chương trình Java bạn sẽ có một tập tin lớp(file class) và bạn không thể chạy nó bằng cách nhấp đôi vào nó được mà bạn phải dùng trình thông dịch(Interpreter) Java để chạy. Mục tiêu của các nhà thiết kế Java là cho phép người lập trình viết chương trình một lần nhưng có thể chạy trên bất cứ phần cứng cụ thể. Ngày nay, Java được sử dụng rộng rãi để viết chương trình chạy trên Internet. Nó là ngôn ngữ lập trình hướng đối tượng độc lập thiết bị, không phụ thuộc vào hệ điều hành.
- + Java chia làm ba bộ phận: J2SE, J2EE, J2ME gồm các đặc tả, công cụ, API để phát triển các ứng dụng.

- + Java có các đặc trưng: đơn giản, hướng đối tượng, độc lập với phần cứng và hệ điều hành, bảo mật, đa luồng, phân tán, động. Vì vậy, Java là một ngôn ngữ được rất nhiều người sử dụng hiện nay.
- + Các kiểu chương trình Java:
  - Applets: Đây là chương trình chạy trên Internet thông qua các trình duyệt hỗ trợ Java như IE hay Netscape. Bạn có thể dùng các công cụ của Java để xây dựng Applet. Applet được nhúng bên trong trang Web hoặc tập tin HTML. Khi trang Web hiển thị trong trình duyệt, Applet sẽ được nạp và thực thi.
  - Ứng dụng thực thi qua dòng lệnh: Các chương trình này chạy từ dấu nhắc lệnh và không sử dụng giao diện đồ họa. Các thông tin nhập xuất được thể hiện tại dấu nhắc lệnh.
  - Ứng dụng đồ họa: Đây là các chương trình Java chạy độc lập cho phép người dùng tương tác qua giao diện đồ họa.
  - Servlet: Java thích hợp để phát triển ứng dụng nhiều lớp. Applet là chương trình đồ họa chạy trên trình duyệt tại máy trạm. Ở các ứng dụng Web, máy trạm gửi yêu cầu tới máy chủ. Máy chủ xử lý và gửi ngược kết quả trở lại máy trạm. Các chương trình Java API chạy trên máy chủ giám sát các quá trình tại máy chủ và trả lời các yêu cầu của máy trạm. Các chương trình Java API chạy trên máy chủ này mở rộng khả năng của các ứng dụng Java API chuẩn. Các ứng dụng trên máy chủ này được gọi là các Servlet. hoặc Applet tại máy chủ. Các xử lý trên Form của HTML là cách sử dụng đơn giản nhất của Servlet. Chúng còn có thể được dùng để xử lý dữ liệu, thực thi các transaction và thường được thực thi qua máy chủ Web.
  - Ứng dụng cơ sở dữ liệu: Các ứng dụng này sử dụng JDBC API để kết nối tới cơ sở dữ liệu. Chúng có thể là Applet hay ứng dụng, nhưng Applet bị giới hạn bởi tính bảo mật.
- + Máy ảo Java (JVM- Java Virtual Machine ): Máy ảo là một phần mềm dựa trên cơ sở máy tính ảo. Nó có tập hợp các lệnh logic để xác định các hoạt động của máy tính. Người ta có thể xem nó như một hệ điều hành thu nhỏ. Nó thiết lập các lớp trừu tượng cho: Phần cứng bên dưới, hệ điều hành, mã đã biên dịch. Trình biên dịch chuyển mã nguồn thành tập các lệnh của máy ảo mà không phụ thuộc vào phần cứng cụ thể. Trình thông dịch trên mỗi máy sẽ chuyển tập lệnh này thành chương trình thực thi.
- + Bộ công cụ phát triển JDK (Java Development Kit) bao gồm: Java Plug-In, chúng cho phép chạy trực tiếp Java Applet hay JavaBean bằng cách dùng JRE thay cho sử dụng môi trường thực thi mặc định của trình duyệt. JDK chứa các công cụ:



- Trình biên dịch javac
  - Trình thông dịch java
  - Trình dịch ngược javap
  - Công cụ sinh tài liệu javadoc
  - Chương trình tìm lỗi (debug) jdb
  - + Môi trường phát triển tích hợp - Integrated Development Environment(IDE):
    - IDE là một loại phần mềm máy tính có công dụng giúp đỡ các lập trình viên trong việc phát triển phần mềm.
    - IDE thường bao gồm:
      - Một trình soạn thảo mã (source code editor): dùng để viết mã.
      - Trình biên dịch (compiler) và/hoặc trình thông dịch (interpreter).
      - Công cụ xây dựng tự động: khi sử dụng sẽ biên dịch (hoặc thông dịch) mã nguồn, thực hiện liên kết (linking), và có thể chạy chương trình một cách tự động.
      - Trình gỡ lỗi (debugger): hỗ trợ dò tìm lỗi.
      - Ngoài ra, còn có thể bao gồm hệ thống quản lí phiên bản và các công cụ nhằm đơn giản hóa công việc xây dựng giao diện người dùng đồ họa (GUI).
      - Nhiều môi trường phát triển hợp nhất hiện đại còn tích hợp trình duyệt lớp (class browser), trình quản lí đối tượng (object inspector), lược đồ phân cấp lớp (class hierarchy diagram),... để sử dụng trong việc phát triển phần mềm theo hướng đối tượng.
    - Một số IDE như: Eclipse, NetBean...
  - + Để bắt đầu phát triển ứng dụng Java ta cần chuẩn bị môi trường lập trình: cài đặt JDK, IDE, thiết lập biến môi trường. Cũng như tìm hiểu kiến thức về lập trình hướng đối tượng. Java là ngôn ngữ phổ biến hiện nay, nên tài liệu hỗ trợ cho người lập trình cũng rất đa dạng và đầy đủ.
- Adobe AIR:
- + AIR (Adobe Integrated Runtime) là 1 môi trường runtime đa nền cho phép lập trình viên có thể sử dụng các kĩ năng lập trình web vốn có của mình như HTML, AJAX, Flash, Flex để xây dựng và triển khai các ứng dụng Internet đẹp, mạnh mẽ, nhiều tính năng (RIAs) ngay trên desktop.
  - + Adobe AIR cho phép các nhà phát triển để sử dụng các công cụ quen thuộc hoặc bất kỳ trình soạn thảo văn bản để xây dựng các ứng dụng của họ và tạo ra 1 ứng dụng cài đặt duy nhất làm việc trên hệ điều hành. Với AIR, chúng ta sẽ chứng kiến một làn sóng các ứng dụng lai (giữa Web với desktop). AIR sẽ bổ sung cho các ứng dụng Web nhiều tính năng vốn là đặc trưng của desktop như kéo-thả, nhận dạng hệ thống và truy cập hệ thống tập tin

nội bộ. Với AIR, các nội dung flash, HTML và tập tin PDF có thể được "đóng gói" vào trong một ứng dụng duy nhất.

— Action Script:

- + ActionScript là ngôn ngữ lập trình cho Adobe Flash Player và Adobe AIR™ chạy trong môi trường thực. Nó cho phép tương tác, xử lý dữ liệu, và nhiều hơn nữa trong Flash, Flex, và nội dung và các ứng dụng AIR.
- + ActionScript được thực hiện bởi ActionScript Virtual Machine (AVM), là một phần của Flash Player và AIR. Mã ActionScript thường được biên dịch vào định dạng bytecode (một ngôn ngữ lập trình viết và được hiểu bởi máy tính) bằng một trình biên dịch, chẳng hạn như một trong những trình biên dịch được xây dựng vào Adobe Flash CS4 Professional hoặc Adobe Flex™ Builder™, hoặc đó là có sẵn trong Adobe Flex™ SDK. Bytecode này được nhúng trong tập tin SWF, được thực hiện bởi Flash Player và AIR.
- + ActionScript 3.0 cung cấp một mô hình lập trình mạnh mẽ mà sẽ được quen thuộc với các nhà phát triển với một kiến trúc cơ bản về lập trình hướng đối tượng. Phiên bản ActionScript 3.0 có nhiều cải tiến so với các phiên bản trước, nó cung cấp thêm một số tính năng và được thiết kế để tạo thuận lợi cho việc tạo các ứng dụng phức tạp có bộ dữ liệu lớn và hướng đối tượng, cơ sở tái sử dụng mã, thực thi nhanh hơn nhờ sự thừa kế mã trong ActionScript.

## IV. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

### 1. *Mô hình tổng quan*

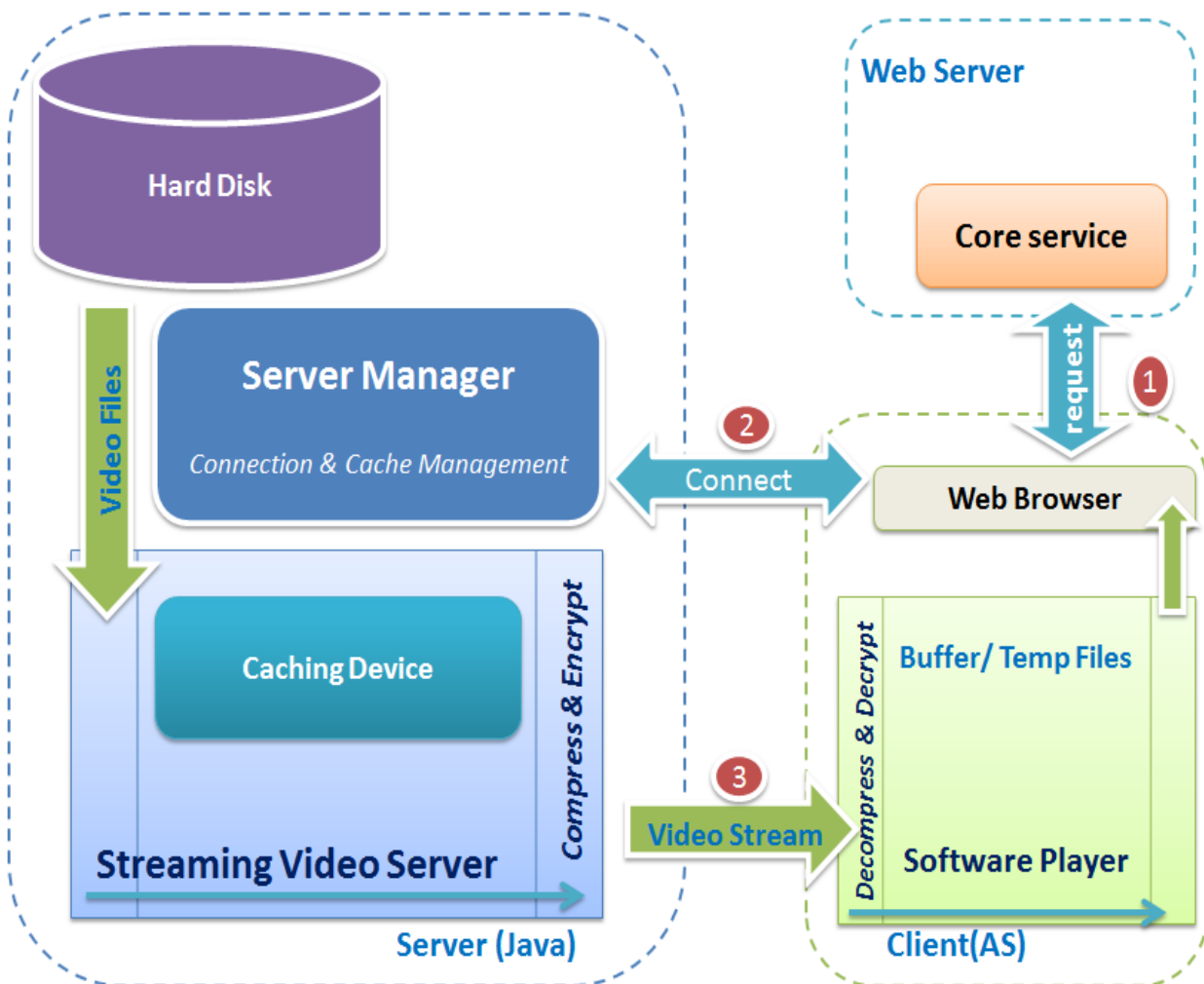
- Hiện nay, chúng ta có rất nhiều hệ thống Website hỗ trợ những bài giảng trực tuyến. Elearning là một trong những ứng dụng khá điển hình đã được xây dựng và vận hành trên Internet. Với mô hình đào tạo này, dữ liệu bài giảng trực tuyến thường là Slide, Video bài giảng, hướng dẫn được lưu trữ trên server, thông tin về bài giảng được lưu trữ trong cơ sở dữ liệu và được hiển thị theo dạng danh sách trên Website đào tạo trực tuyến.
- Người sử dụng truy cập vào Website, lựa chọn bài giảng trong danh sách các bài giảng đó và bắt đầu học tập theo chương trình đào tạo trực tuyến.

### 2. *Xác lập tính năng hệ thống*

Như đã giới thiệu ở trên, tầm vực của đề tài không phải là thiết kế toàn bộ hệ thống trình chiếu Video, cũng như các chức năng quản lý cơ sở dữ liệu và quản lý chức năng của người dùng mà chỉ tập trung vào các chức năng sau:

- Những chức năng quản lý kết nối, quản lý tiến trình Streaming Video, quản lý Cache trên một Server có hỗ trợ Video caching để tăng tốc độ truyền tải Video đến người dùng.
- Phát triển một Core Service để tích hợp vào những Website đào tạo trực tuyến (hay bất kỳ Website cần hỗ trợ trình chiếu Video trực tuyến nói chung). Những website đó mua không gian lưu trữ dữ liệu trên Server của chúng ta, khi đó người dùng của họ sẽ kết nối trực tiếp đến Server, Server truyền tải Video đến người dùng thông qua những kênh Stream.
- Với việc lựa chọn những công nghệ phát triển khác nhau, tùy thuộc vào player hỗ trợ ta có thể phát triển software hoặc plug-in để quản lý dữ liệu bên phía người dùng.

### 3. Kiến trúc hệ thống



Hình 1: Kiến trúc hệ thống

Hệ thống gồm 3 thành phần :

— Web server(W):

- + Web Server là máy chủ có dung lượng lớn, tốc độ cao, là nơi lưu trữ thông tin về các website có cung cấp dịch vụ chia sẻ video trực tuyến (những Website này nhúng phần mã embeded flash player (do nhà cung cấp dịch vụ lưu trữ và quản lý tập tin(S) cung cấp) cùng với những thông tin liên quan khác).
- + Khi Web Server nhận yêu cầu từ Client về việc xem một video có trong danh sách video mà Website cung cấp, thì nó sẽ gọi về Client: đoạn mã html tĩnh chứa tập tin embeded flash và tập tin xml chứa thông tin của video đó.(2')

— Server(S):

- + Server là một máy chủ gồm có: Hard disk (D) chứa dữ liệu, và thành phần quản lý máy chủ(SM) quản lý quá trình kết nối của các máy trạm (thông qua một cổng nhất định).
- + Hard Disk(D): Lưu trữ tập tin video của nhà cung cấp.
- + Server Manager(SM):
  - o Tạo Server với port được lựa chọn, chờ người dùng kết nối tới.
  - o Quản lý người dùng kết nối dựa trên IP.
  - o Quản lý những tập tin video đã được đưa vào cache.
  - o Quản lý việc lựa chọn gửi phần nào của tập tin video về cho người dùng.
- + Streaming Video Server (SV):
  - o Khi nhận được yêu cầu từ phía người dùng lấy một tập tin video, Streaming Video Server tìm tập tin đó dưới ổ cứng(D) và đẩy vào Caching Device.
  - o Tại Caching Device, tập tin video được cắt ra thành nhiều phần (segment) nhỏ lưu trữ dưới dạng một cấu trúc cây Binary Tree.
  - o Sau đó từng phần của tập tin video sẽ được đưa qua một cơ chế nén và mã hóa trước khi gửi về người dùng.(3)

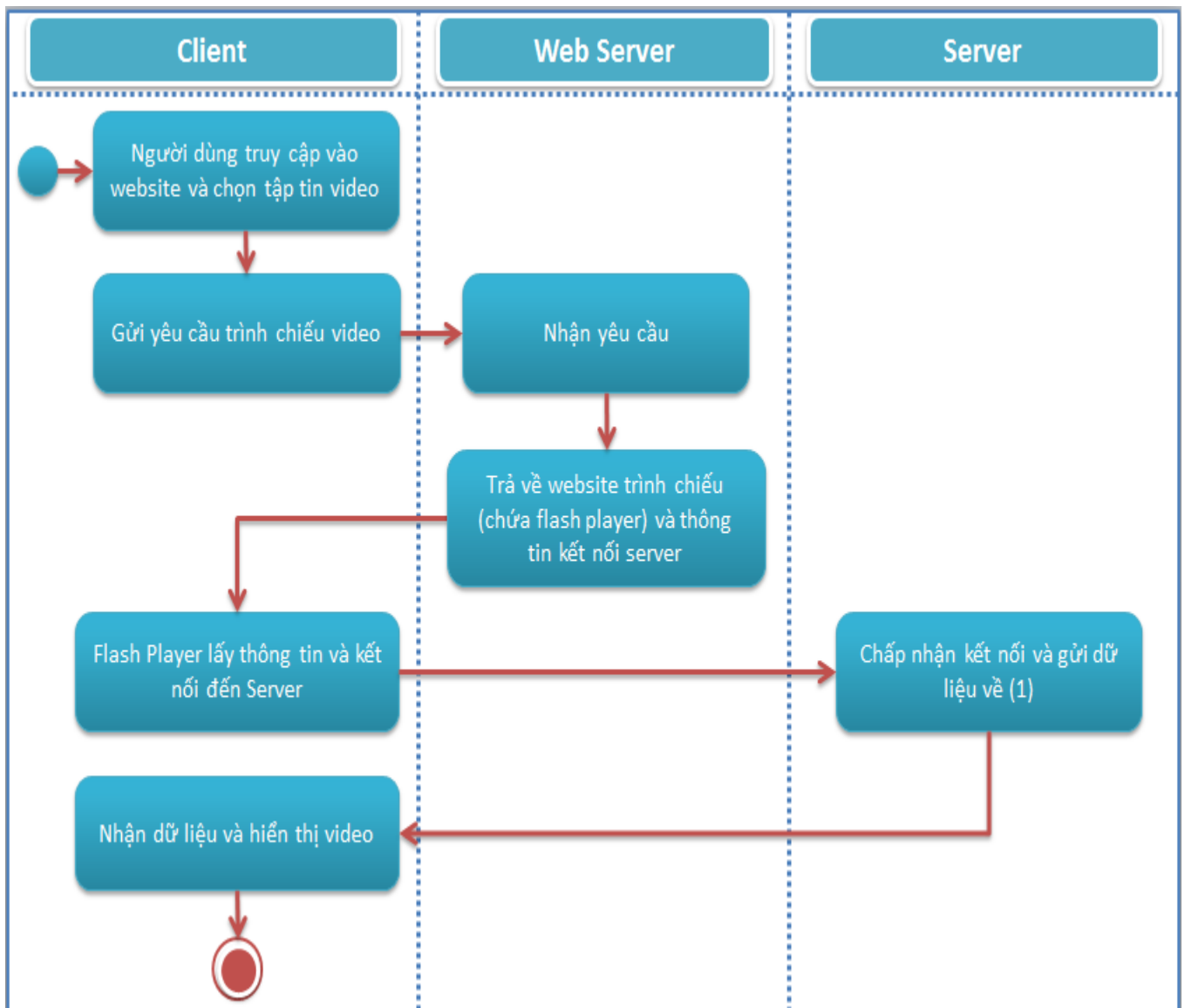
— Client(C):

- + Truy cập vào web browser, chọn tập tin video cần xem và bấm play, lệnh yêu cầu play tập tin video sẽ được gửi lên Web Server.(1)
- + Web browser sẽ nhận software player (trong đoạn mã html tĩnh chứa tập tin embeded flash) và thực hiện kết nối với server.(2)
- + Software player nhận những video stream (3') và thực hiện việc giải nén, giải mã trước khi hiển thị Video lên cho người dùng.

Tóm lại:

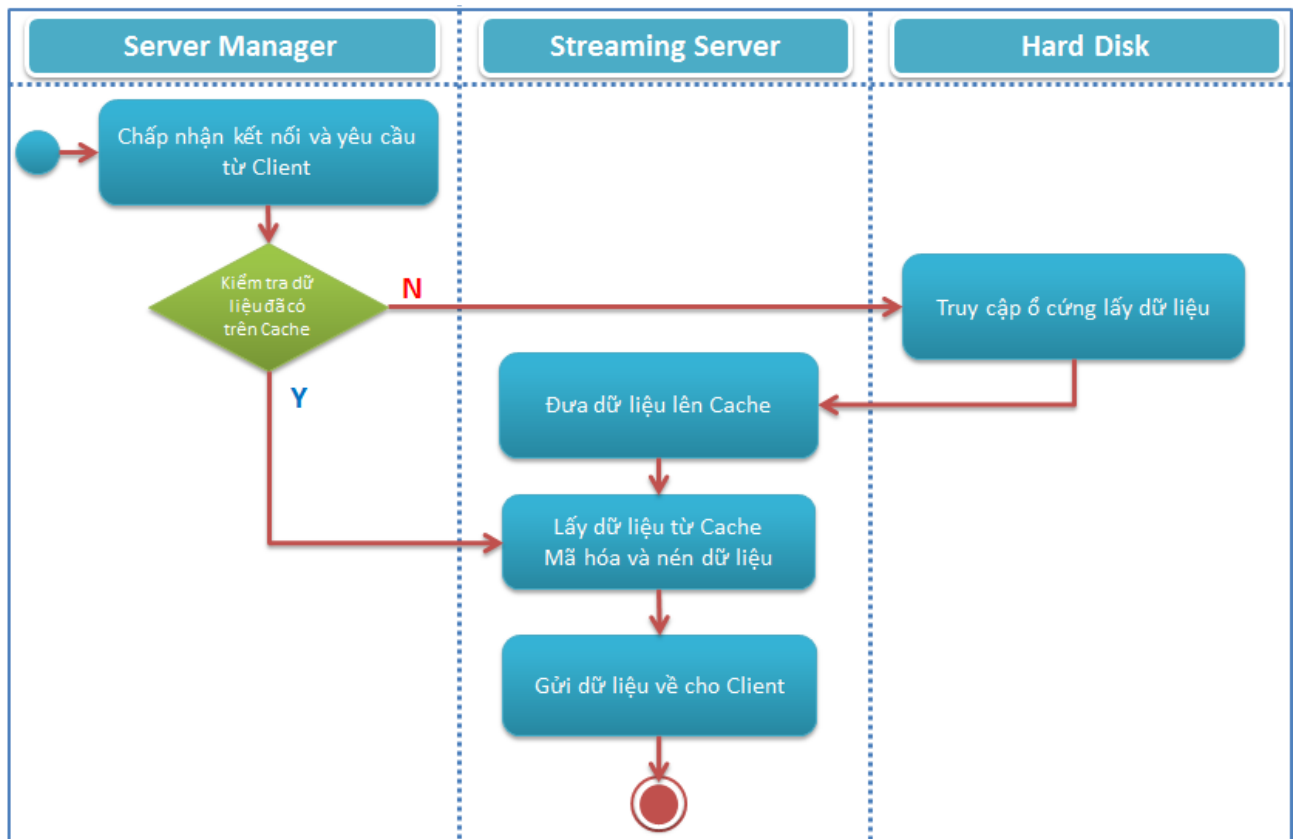
- Web Server: trả về người dùng đoạn html tĩnh, nội dung tập tin chứa embedded flash.
- Server: lưu trữ tập tin video, gửi tập tin video được yêu cầu về người dùng.
- Client: nhận thông tin từ Web Server và kết nối đến server ( với thông tin nhận được từ Web Server tương ứng về tập tin video đã chọn).

- **Mô hình hoạt động tổng quát của hệ thống:**

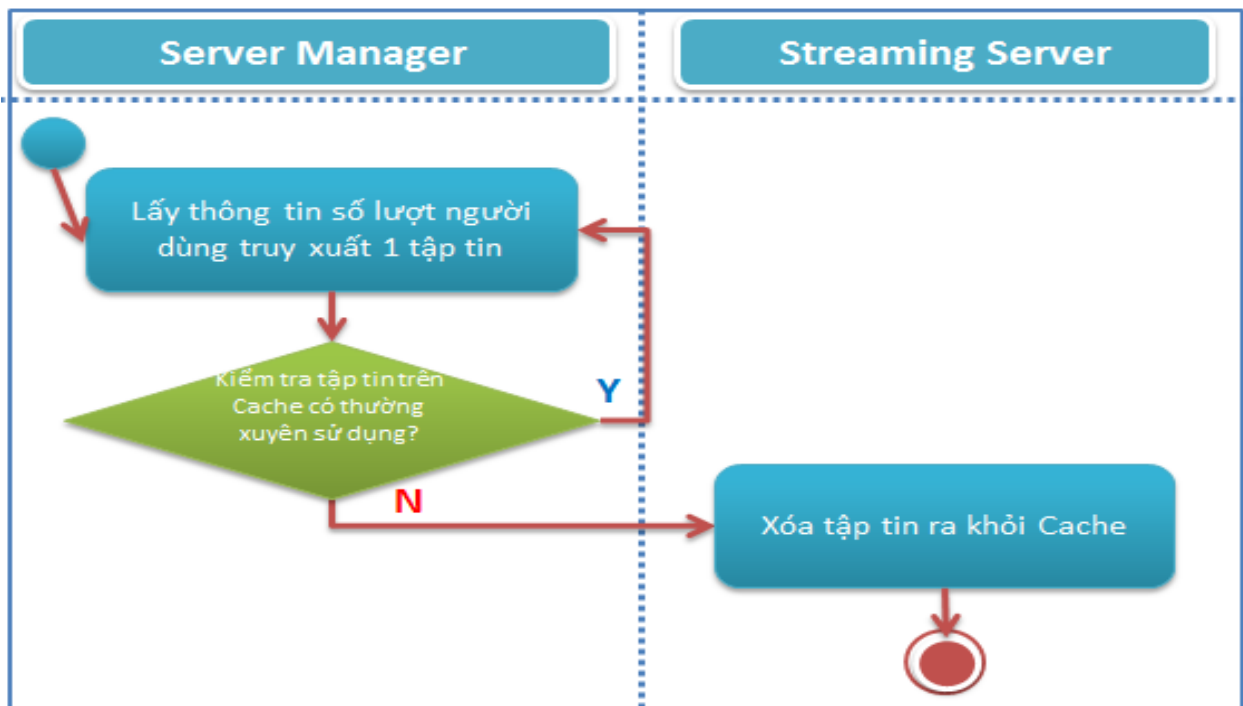


Hình 2: Lược đồ hoạt động tổng quát

— Từ (1) ta có mô hình hoạt động chi tiết bên phía Server



Hình 3: Lược đồ hoạt động trên Server

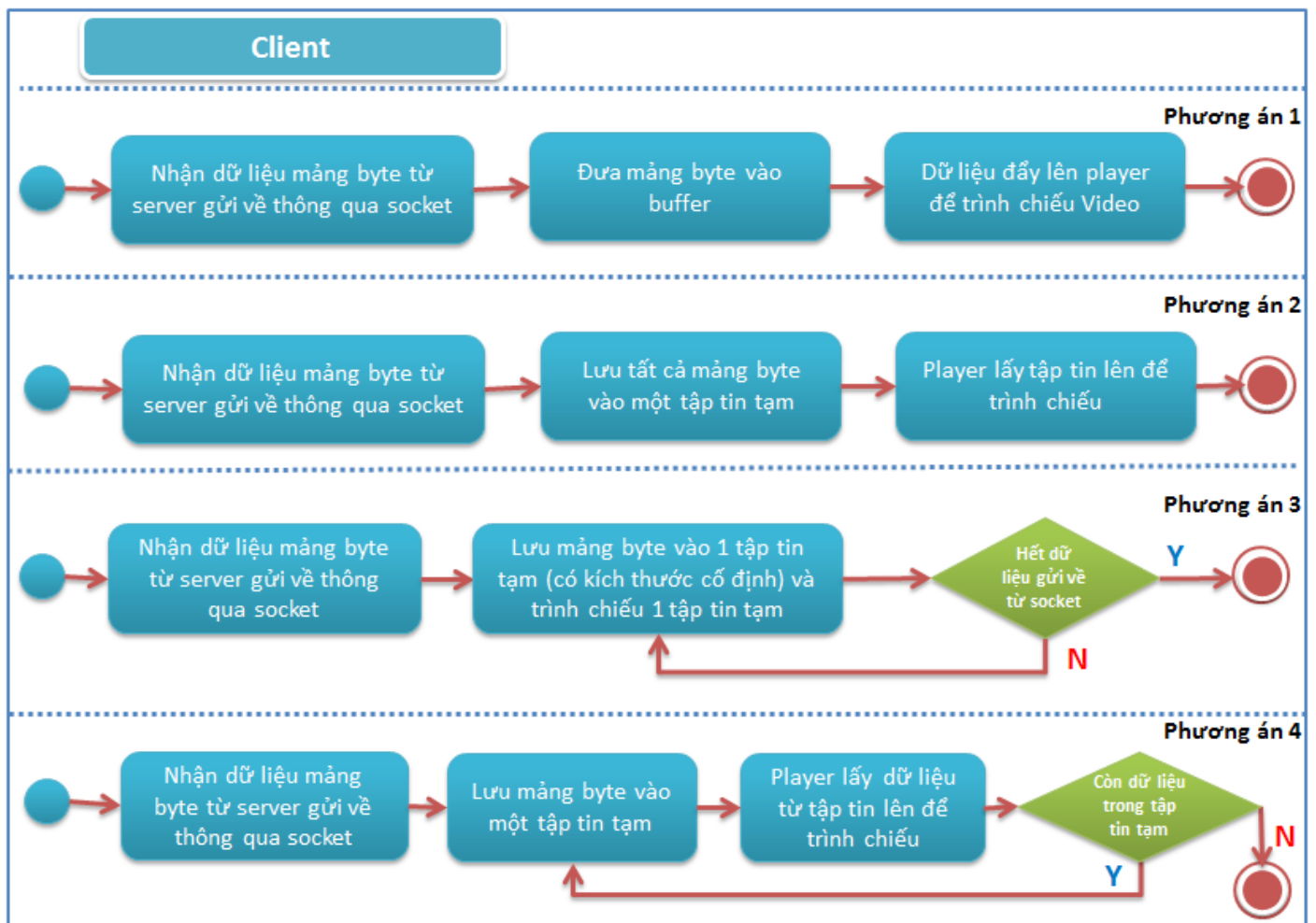


Hình 4: Lược đồ hoạt động quản lý Cache trên Streaming Server

Đề xuất giải pháp phát triển hệ thống:

Trong giai đoạn đồ án 2, nhóm giải quyết xử lý bên phía Client theo một số phương án đề ra như sau(Sau khi nhận dữ liệu từ socket về và lưu vào những mảng byte):

- Phương án (1): Viết lại module xử lý quá trình hiển thị một frame. Tiến hành đọc các mảng byte đó theo trình tự và đưa dữ liệu vào buffer, để từ buffer hiển thị nội dung lên player.
- Phương án (2): Do tính bao đóng của thư viện NetStream trong actionscript, nên không can thiệp được vào quá trình phát một tập tin Video. Vì vậy, tiến trình phát tập tin phải đợi đến khi kết thúc tiến trình nhận tập tin; sau đó lưu tất cả mảng byte xuống một tập tin tạm của hệ thống; player sẽ lấy tập tin lên và phát tập tin tạm đó.
- Phương án (3): tương tự phương án (2) nhưng ta lưu trữ dữ liệu vào hai tập tin (có thể nhiều hơn) để chia quá trình xử lý
  - + Quá trình ghi dữ liệu.
  - + Một tập tin dành cho quá trình phát Video.
- Phương án (4): tương tự phương án (2) nhưng ta can thiệp vào quá trình ghi dữ liệu vào tập tin tạm, và phát những dữ liệu đã ghi được.





*Hình 5: Lược đồ một số giải pháp phát triển hệ thống*

## V. THỰC HIỆN

### 1. **Server:**

#### a. *Chờ kết nối :*

- Tập tin thực hiện : Server.java
- Tập tin liên kết : MiniServerDlg.java, ServerThread.java
- Thực hiện chức năng chờ client kết nối đến và chấp nhận kết nối. Sau đó tạo ra mỗi thread tương ứng với mỗi client kết nối đến.

#### b. *Đa luồng:*

- Tập tin thực hiện : ServerThread.java
- Tập tin liên kết: Server.java, ReadFile.java
- Thực hiện chức năng đa luồng trên server, nhằm đáp ứng số lượng kết nối lớn tới server. Mỗi thread ứng với mỗi kết nối client đến server.
- Nhận thông tin về tập tin cần xem từ người dùng.
- Gửi dữ liệu về cho người dùng.

#### c. *Đọc tập tin:*

- Tập tin thực hiện: ReadFile.java
- Tập tin liên kết: ServerThread.java
- Thực hiện chức năng đọc tập tin video từ đĩa cứng lên bộ đệm.

### 2. **Client:**

#### a. *Kết nối server:*

- Tập tin thực hiện: Connection.as
- Tập tin liên kết: myVideo.as
- Thực hiện chức năng kết nối với server.
- Gửi thông tin về tập tin video cần xem lên server.
- Nhận dữ liệu từ socket và lưu thành tập tin tạm của hệ thống.

#### b. *Tải video:*

- Tập tin thực hiện: VideoLoader.as
- Tập tin liên kết: myVideo.as
- Thực hiện chức năng tải video từ tập tin tạm lên bộ đệm trước khi phát.

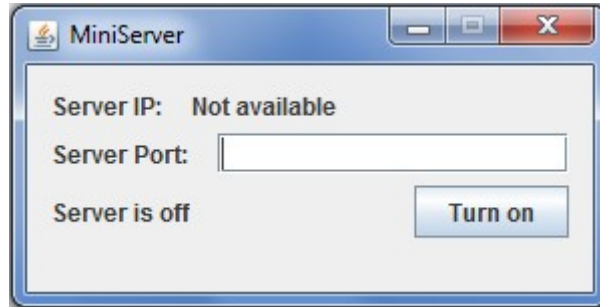
#### c. *Player:*

- Tập tin thực hiện: myVideo.as
- Tập tin liên kết: VideoLoader.as, Connection.as
- Thực hiện các chức năng của player như phát, dừng, phát lại tập tin Video; tăng, giảm âm thanh.

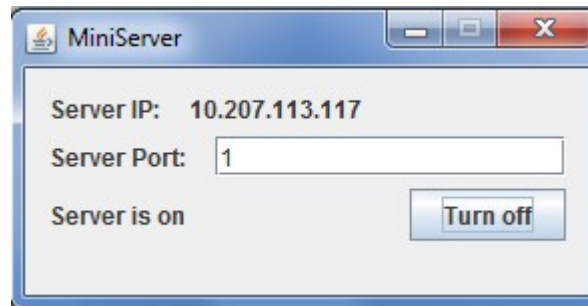
### 3. **Giao diện:**

*a. Server:*

- Khi khởi động chương trình dành cho server thì khung cửa sổ của MiniServer sẽ hiện ra cho người quản trị server chọn port để chờ kết nối.

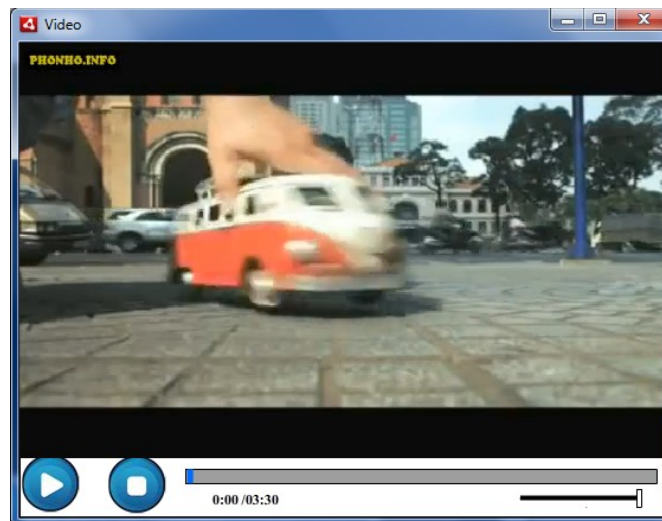


- Sau khi Turn On thì server sẽ bắt đầu chạy và chờ client kết nối đến.



*b. Client:*

- Sau khi kết nối với server, client sẽ nhận dữ liệu về tập tin video và phát cho người dùng xem.



## VI. TỔNG KẾT ĐỒ ÁN

- Trong giai đoạn đồ án đã tìm hiểu được những kiến thức liên quan, và thiết kế một hệ thống gồm có:
  - + Server: xây dựng một mini server bằng java
    - Mở 1 socket với port do mình lựa chọn và chờ người dùng connect đến.
    - Server sẽ tạo ra một thread tương ứng với mỗi người dùng connect đến.
    - Server quản lý những người dùng đã được chấp nhận kết nối dựa trên ip của nó.
    - Đọc tập tin video từ ổ cứng đưa lên buffer và thực hiện việc cắt tập tin thành nhiều tập tin nhỏ trước khi gửi về người dùng.
    - Server sẽ đóng connection sau khi hoàn thành việc gửi tập tin về người dùng.
  - + Client: Xây dựng công cụ flash player nhúng vào website bằng actionscript.
    - Client connect đến server với ip và port của server được cung cấp.
    - Gửi tên tập tin cần xem lên server.
    - Nhận tập tin về từ socket và lưu trong mảng byte.
    - Lưu trữ mảng byte thành tập tin temp của hệ thống và tải nó lên flash player hiển thị cho người dùng xem.
- Bên cạnh đó, nhóm còn gặp những vấn đề khó khăn chưa giải quyết được bên phía Client:
  - + Với phương án (1): vì vấn đề bao đóng của thư viện Video trong ActionScript nên không can thiệp được vào quá trình đọc dữ liệu từ mảng byte đó đưa vào một buffer để hiển thị lên trên player.
  - + Với phương án (2): Phải đợi quá trình tải tập tin về hoàn thành thì chương trình mới được lấy tập tin lên để phát.
  - + Với phương án (3): Nếu lưu những mảng byte đó xuống hai (hay nhiều hơn) tập tin tạm của hệ thống: một tập tin để ghi, một tập tin để phát. (hai tập tin này được dùng lại liên tục sau khi tiến trình hoàn thành). Hai tiến trình này xử lý song song, chưa giải quyết được vấn đề chờ sự kiện hoàn tất việc đọc/ ghi trên 2 tập tin:
    - Tiến trình ghi vào một tập tin có thể ghi nhanh hơn tiến trình phát, khi đó tiến trình ghi phải đợi tiến trình phát thực hiện xong thì mới được phép ghi lên tập tin còn lại.
    - Ngược lại, tiến trình phát thực hiện phát một tập tin xong trước phải đợi tiến trình ghi thực hiện xong, thì mới phát tập tin còn lại được, khi đó hình ảnh sẽ dừng lại.
  - + Với phương án (4): Nếu thực hiện vừa ghi vừa phát trên một tập tin tạm thì hệ điều hành sẽ ko cho phép, báo lỗi vi phạm và ngừng ngay ứng dụng.

## VII. PHƯƠNG HƯỚNG PHÁT TRIỂN LUẬN VĂN

### 1. *Nghiên cứu*

#### a. *Quản lý Video Caching*

Nhiệm vụ chính trong giai đoạn luận văn là nhóm sẽ xây dựng một Cache hỗ trợ Video Caching bên phía Server. Để quản lý Cache cho hiệu quả, những kiến thức cần nghiên cứu và thử nghiệm gồm có:

- Thống kê và lưu trữ thông tin gửi yêu cầu từ phía người dùng kết nối đến Server. Phân tích thông tin, xử lý dữ liệu bằng cách sử dụng công cụ khai phá dữ liệu là gom cụm dữ liệu để được số liệu thống kê những tập tin thường xuyên được người dùng yêu cầu nhất.
- Nghiên cứu thuật toán quản lý Cache, giải quyết bài toán đưa tập tin vào Cache, cũng như loại bỏ ra khỏi Cache cho hợp lý.
- Sử dụng những dữ liệu thống kê viết testcase để kiểm thử giải thuật.

#### b. *Quản lý Streaming*

- Chuẩn bị nội dung (dữ liệu truyền tải):
  - + Nội dung video thô (chẳng hạn như những hình ảnh sống đc tạo ra bởi 1 camera hoặc video đã đc ghi vào một đoạn băng...) không thích hợp với ứng dụng streaming. Thông thường, nội dung cần đc xử lý, có thể bao gồm chuyển đổi định dạng, nén video, thêm nhãn và chỉ mục và xuất bản cho streaming.
  - + Chuẩn bị nội dung cho việc xem có thể là một quá trình đơn giản hoặc rất phức tạp, phụ thuộc trên kết quả người sử dụng và khả năng đáp ứng của họ về thời gian và tiền bạc. Ở đây chúng ta quan tâm số lượng những xử lý cần phải được thực hiện để chuẩn bị nội dung cho streaming.
- + Mã hóa và giải mã:
  - Mã hóa là cách thức chuyển đổi thông tin từ dạng clear text sang dạng cipher text hoặc ngược lại, nhằm chống lại những truy cập bất hợp pháp tới dữ liệu được tuyên đi trên mạng. Mục đích của mã hóa là che dấu nội dung thông tin, đảm bảo dữ liệu truyền đi được an toàn, đảm bảo cho người nhận dữ liệu xác thực được dữ liệu nhận là dữ liệu gốc, không có kẻ giả mạo, hoặc thay đổi dữ liệu, cũng như người nhận hoặc người gửi không thể chối bỏ sau khi gửi hoặc nhận thông tin.

- Trong quá trình chuẩn bị dữ liệu, một số hệ thống sử dụng một vài phần mềm phi thời gian thực dịch mã sử dụng mã hóa two-pass. Có nghĩa là mã hóa video một lần, đánh giá, và sau đó mã hóa thêm một lần nữa để cho ra kết quả cuối cùng. Lợi ích của hệ thống two-pass là kết quả của lần mã hóa đầu tiên có thể được sử dụng để điều khiển làm thế nào để mã hóa lần hai được thực hiện. Mặt hạn chế của mã hóa two-pass là nó yêu cầu nhiều xử lý hơn mã hóa single-pass, điều này là một vấn đề không bình thường cho nội dung được lưu trữ trên server và phát lại sau đó.
- Trong giai đoạn luận văn, nhóm sẽ nghiên cứu sâu hơn về việc có nên áp dụng phương thức mã hóa(giải mã) thông tin trước (sau) khi truyền (nhận) dữ liệu. Nếu có sẽ chọn lựa và thử nghiệm giải thuật mã hóa hợp lý.
- + Nén và giải nén:
  - Trong kỹ thuật truyền tin nối tiếp, do các bit dữ liệu được truyền đi nối tiếp, kênh truyền bị giới hạn về băng thông và các chuẩn ghép nối, nên tốc độ truyền tin tương đối thấp. Để tăng tốc độ truyền tin ta có thể sử dụng nhiều phương pháp như kỹ thuật điều chế pha nhiều mức, điều chế QAM, TCM... Nén dữ liệu trước khi truyền đi cũng là một trong các phương pháp nhằm tăng tốc độ truyền dữ liệu. Một phương pháp thực hiện nén các tập tin ngay tại máy chủ trước khi truyền đi, tại các máy tính nhận, các tập tin lại được giải nén để phục hồi lại dạng ban đầu. Ưu điểm là bên phát và bên thu chỉ cần có chung phần mềm nén và giải nén, ngoài ra còn có thể áp dụng được để truyền dữ liệu qua các modem không hỗ trợ nén dữ liệu hoặc truyền dữ liệu trực tiếp qua cổng COM của máy tính. Nhược điểm của phương pháp này là các máy vi tính phải tốn thêm thời gian nén và giải nén, nhưng do sự phát triển nhanh chóng của các bộ vi xử lý mà thời gian thực hiện nén và giải nén được giảm nhỏ hơn rất nhiều thời gian để truyền dữ liệu.
  - Thông thường, hầu hết các tập tin có rất nhiều thông tin dư thừa, việc thực hiện nén tập tin thực chất là mã hoá lại các tập tin để loại bỏ các thông tin dư thừa. Trong nhiều hệ thống chỉnh sửa, nội dung được chuyển đổi về một định dạng phổ biến trước khi việc xử lý diễn ra.
  - Một định dạng phổ biến thường được sử dụng là 32-bit RGB, mỗi pixel của mỗi frame có 8 bit dữ liệu cho mỗi ba màu cơ bản đỏ, xanh lá cây, xanh dương. Định

- dạng này rất dễ làm việc và xuất ra kết quả có chất lượng cao, nhưng nó có hạn chế là yêu cầu khoảng trống chứa dữ liệu lớn (nhiều megabyte cho mỗi giây video). Video từ một định dạng nén phải được giải nén ra định dạng phổ biến đó. Chuẩn bị nội dung tốt sẽ có đủ thời gian để một giải thuật nén phức tạp hoạt động.
- Giai đoạn sau của đề tài, nhóm sẽ nghiên cứu một số phương pháp nén dữ liệu như: Run-Length Encoding, Huffman, phương pháp nén LZW... Và lựa chọn ra phương thức nén thông qua các tiêu chí hệ số nén, mức độ phù hợp với kiểu tập tin truyền nhận, cũng như khả thi trong tốc độ...
  - + Kết xuất tập tin XML
    - Như kiến trúc đề xuất, luồng hoạt động để Người dùng có thể xem được Video: Người dùng sẽ truy cập vào Website đào tạo trực tuyến, lựa chọn một video trong danh sách; khi đó, Web Server sẽ gửi thông tin để kết nối với Server về cho Client, thông thường những thông tin đó gửi qua socket – nên tính bảo mật đường truyền không cao.
    - Nhóm dự định một Core Service tích hợp vào Web Server, cơ chế hoạt động như sau của dịch vụ đó như sau: Web Server sẽ kết xuất thông tin theo định dạng tập tin XML gửi về cho Client. Thông tin trong tập tin XML gồm: thông tin về Server để Client kết nối đến Server và thông tin về tập tin Video được lưu trữ trên Server.
    - Ngoài ra, tác vụ seek bên Client sẽ sinh ra những thay đổi trong thứ tự yêu cầu gói tin; được hiểu như sau: yêu cầu dữ liệu bên phía Client không phải là nhận liên tục một chuỗi các gói tin đã được cắt ra theo thứ tự của tập tin Video gốc, mà lựa chọn gói tin có thứ tự bất kỳ; khi đấy, những yêu cầu từ phía Client gửi đến Server khi đang trong quá trình truyền nhận dữ liệu cũng sẽ được đóng gói thành tập tin XML để gửi đi.
  - Streaming Server: Quản lý luồng gửi tập tin từ Server về cho Client.
  - Nghiên cứu giải pháp thiết lập một buffer bên phía Client và quản lý dữ liệu trên buffer.

## 2. *Hiện thực*

Hiện thực hệ thống theo kiến trúc đã đưa ra. Mục tiêu và tầm vực của luận văn là hiện thực chức năng quản lý Cache trên Server. Có khả năng mở rộng sau khi nghiên cứu thành công sẽ tích hợp các chức năng mã hóa, nén và kết xuất XML.

## VIII. TÀI LIỆU THAM KHẢO

- (1) <http://osflash.org/flv>
- (2) **Video Over IP** – Wes Simpson
- (3) **Using Flash** – Adobe
- (4) [www.java.com](http://www.java.com)
- (5) **An Interactive Video Delivery and Caching System Using Video Summarization** - Sung-Ju Lee, Wei-Ying Ma, and Bo Shen

### NHẬT KÍ LÀM VIỆC ĐỒ ÁN MÔN HỌC 2

Thời gian gặp gỡ	Nội dung trao đổi
Tuần 1, 27/02/2011–04/03/2011	Nhận đồ án môn học và trao đổi công việc
Tuần 2, 07/03/2011–11/03/2011	Tìm hiểu về Media Player, Window ActiveX Thử nghiệm một số công cụ trình chiếu Video của Microsoft
Tuần 3, 14/03/2011-18/03/2011	Tìm hiểu khả năng hỗ trợ Video của HTML5, tìm hiểu ưu nhược điểm của các loại tập tin Video Tìm hiểu qui trình hoạt động của Video Caching
Tuần 4, 21/03/2011- 25/03/2011	Lựa chọn công nghệ để phát triển hệ thống Trình chiếu được Video trên Web browser từ local host

Tuần 5, 28/03/2011–01/04/2011	Tìm hiểu về Streaming Video Server Tự xây dựng Flash Player sử dụng ngôn ngữ ActionScript
Tuần 6, 7	Thi giữa kì
Tuần 8, 18/04/2011–22/04/2011	Tìm hiểu về Progressive download Hoàn thiện chức năng của Flash Player
Tuần 9, 25/04/2011-29/04/2011	Tìm hiểu kiến trúc tập tin FLV Đưa Flash lên Web
Tuần 10, 02/05/2011– 06/05/2011	Xây dựng Server bằng Java Cắt tập tin FLV và lưu trữ ở Server
Tuần 11, 09/05/2011– 13/05/2011	Tìm hiểu cách thức xử lý quá trình liên kết Client- Server Giải quyết bài toán tải dữ liệu về
Tuần 12, 16/05/2011–20/05/2011	Tìm hiểu cách thức đọc dữ liệu từ buffer Nghiên cứu cách thức đọc dữ liệu từ buffer của Flash Player Đưa ra những phương pháp giải quyết bài toán hiện thị dữ liệu phía Client
Tuần 13, 23/05/2011–27/05/2011	Nghiên cứu cách thức đọc dữ liệu từ buffer của Flash Player Thử nghiệm bài toán lưu trữ
Tuần 14, 30/05/2011–03/06/2011	Thử nghiệm giải pháp lưu thành 2 tập tin tạm của hệ thống.
Tuần 15, 06/06/2011 –10/06/2011	Tổng kết, chuẩn bị báo cáo
Tuần 16, 17/06/2011	Nộp báo cáo.



