Tạp chí Khoa học Giáo dục Kỹ thuật, số 9(3/2008)
Đại học Sư phạm Kỹ thuật Thành phố Hồ Chí Minh

ỨNG DỤNG FLASH TRONG E-LEARNING

Trần Anh Khoa
Phạm Thái Bảo
Đào Văn Tuyết

TÓM TẮT

E-Learning hiện nay đang là vấn đề đáng quan tâm của Công nghệ thông tin. E-learning giúp cho học viên ở xa vẫn có thể tiếp cận được với tài liệu mới nhất và được hướng dẫn bởi giáo viên tốt nhất (theo lời của Bill Gates). E-learning giúp học viên giảm được đáng kể chi phí so với các khóa học tập trung, và học viên có thể học tập bất cứ khi nào có thời gian rỗi.

Cùng với sự phát triển của CNTT thì E-learning cũng có nhiều thay đổi. Các công nghệ Video Conference giúp giảng viên và học viên có thể giao tiếp trực tiếp với nhau và giữa các học viên với nhau chứ không chỉ là hình thức download tài liệu về nghiên cứu sau đó upload kết quả và thi từ xa.

Một trong những cách tiếp cận Video Confence là sử dụng Flash. Hơn thế nữa Flash còn hỗ trợ Shared Desktop, WhiteBoard, chat, Record Video ,.. giúp cho tương tác giữa giảng viên với học viên và giữa học viên với nhau được tiện lợi hơn. Một điểm mạnh khác của Flash là có thể nhúng vào trình duyệt Web nên rất dễ dàng tích hợp vào các hệ thống E-learing dựa trên CMS.

Nội dung bài báo cáo gồm 5 phần: 1.Giới thiệu về Flash và Media Server. 2.Giao thức truyền thông giữa Flash Client và Media Server. 3.Xây dựng module Video Conference, Shared Desktop và lưu bài giảng. 4.Một số hình ảnh thử nghiệm. 5.Kết luận và hướng phát triển.

Từ khóa: *Flash, Media Server, Video Conference, Shared Desktop,…*

I. GIỚI THIỆU VỀ FLASH VÀ MEDIA SERVER

Flash trước đây là sản phẩm của công ty Macromedia, hiện nay Flash thuộc sở hữu của công ty Adobe. Nhắc đến Flash thường người ta nghĩ đến những đoạn phim hoạt hình chứ ít ai nghĩ nó có thể thực hiện Video Conference. Nhưng hiện nay Flash đã làm được điều đó. Để thực hiện Video Conference bằng Flash thì cần sử dụng một Media Server (Macromedia Server hay Red5 – một sản phẩm mã nguồn mở). Flash client giao tiếp với Media Server thông qua giao thức RTMP (Real Time Messaging Protocol) sử dụng phương pháp đóng gói thông điệp AMF (ActionScript Message Format). Không những có thể thực hiện Video Conference, Flash còn hỗ trợ Shared Object, Record Video, và còn nhiều tính năng khác nữa… *Các tính năng mà Flash hỗ trợ mạnh đều có thể ứng dụng vào trong E-learning.*

Flash sử dụng chuẩn nén video H.264 nên có hệ số nén cao ngoài ra Flash còn cung cấp khả năng tùy biến chất lượng video rất lớn nên có thể hoạt động trên nhiều kênh truyền từ tốc độ thấp đến tốc độ cao.

II. GIAO THỨC TRUYỀN TẢI THỜI GIAN THỰC RTMP (REAL TIME MESSAGING PROTOCOL) VÀ PHƯƠNG PHÁP ĐÓNG GÓI THÔNG ĐIỆP AMF (ACTIONSCRIPT MESSAGE FORMAT)

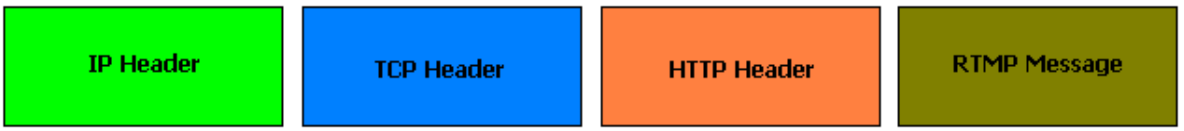
1. Giao thức truyền tải thời gian thực RTMP

1.1. Cơ bản về RTMP

RTMP là một giao thức đơn giản và hiệu quả, được tối ưu cho việc sử dụng băng thông. Nó có thể hỗ trợ tối đa 64 luồng dữ liệu trên cùng một kết nối. Trong header của một AMF có chứa chỉ số của luồng dữ liệu mà nó thuộc về. Một message RTMP có thể chứa nhiều hơn một đối tượng AMF.

RTMP ở chế độ tiêu chuẩn chạy trên TCP với cổng mặc định là 1935. Ngoài ra RTMP có thể chạy trong chế độ đường hầm trên một kết nối HTTP sử dụng cổng 80.

Hình 1. *RTMP ở chế độ tiêu chuẩn*



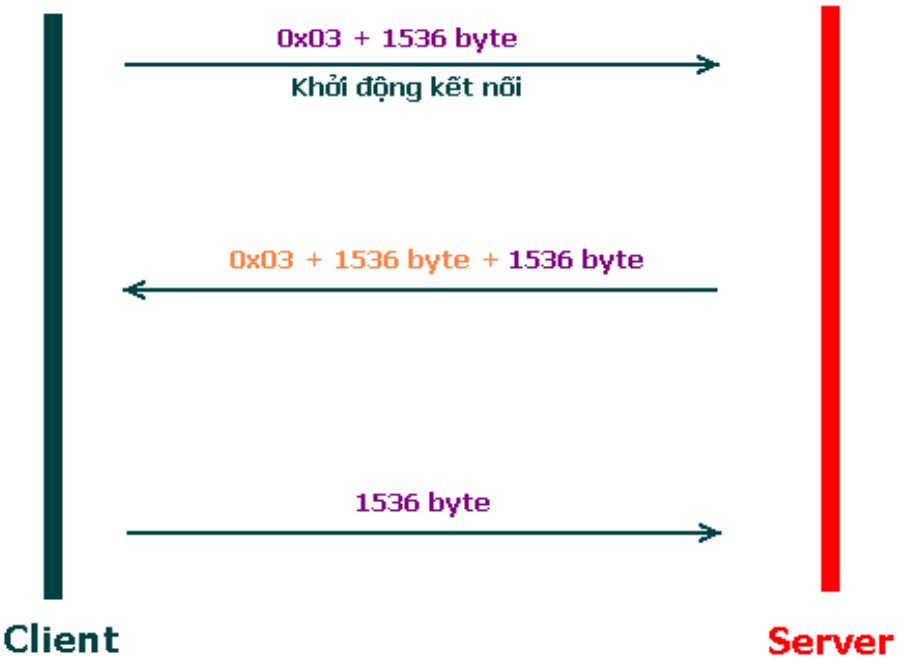
Hình 2. *RTMP ở chế độ đường hầm*

1.2. Quá trình bắt tay

Hoạt động cơ bản của RTMP như sau : Tất cả quá trình truyền thông được khởi động bởi client. Client khởi tạo một kết nối RTMP bằng cách gửi một byte có giá trị 0x03 – byte này được theo sau bởi một khối dữ liệu 1536 byte. Định dạng của khối dữ liệu này cho đến nay vẫn chưa biết nhưng nó dường như không thực sự được sử dụng bởi giao thức ngoại trừ thao tác bắt tay.

Server khi nhận được gói dữ liệu sẽ lưu lại khối dữ liệu 1536 byte này, và cũng gởi 1 byte giá trị 0x03 theo sau bởi hai khối 1536 byte. Khối thứ hai chính là nội dung đã được gửi lên bởi client trước đó.

Client nhận hai khối dữ liệu 1536 byte từ server, so sánh với khối dữ liệu ban đầu nó gửi lên server, nếu phù hợp thì kết nối được thiết lập, nó cũng gửi trả khối dữ liệu này về lại cho server.



Hình 3. *Quá trình bắt tay giữa Client và Server trong giao thức RTMP*

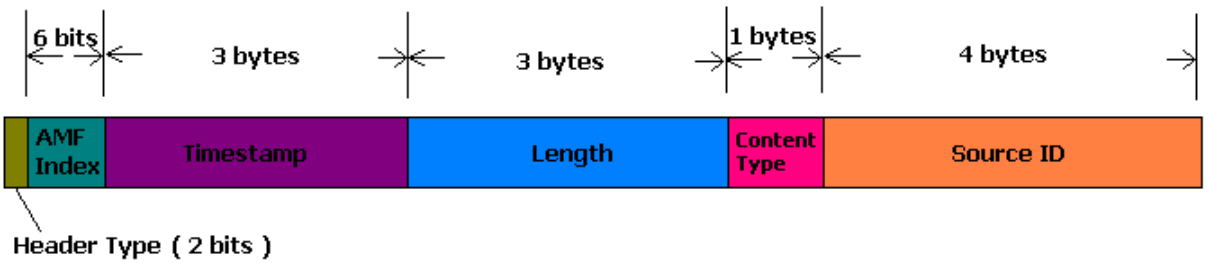
Sau thao tác bắt tay Client tiếp tục gửi ba đối tượng AMF lên server để khởi động truyền dữ liệu. Đối tượng đầu tiên là đối tượng connect nhằm thông báo client đã sẵn sàng cho quá trình truyền thông. Đối tượng AMF thứ hai chính là đối tượng NetConnection từ client, lớp Action Script này được sử dụng tạo kết nối tới media server. Đối tượng AMF thứ ba là đối tượng NetStream từ client dùng để xác định file cần stream từ server.

1.3. Tiêu đề RTMP

RTMP có bốn loại tiêu đề đó là tiêu đề 12, 8, 4 hoặc 1 byte. Byte đầu tiên của tiêu đề rất quan trọng, 2 bit đầu tiên của nó xác định kích thước của tiêu đề.

- 0x00: tiêu đề 12 byte
- 0x01: tiêu đề 8 byte
- 0x02: tiêu đề 4 byte
- 0x03: tiêu đề 1 byte

Sáu bít còn lại biểu diễn chỉ số của đối tượng AMF. Một khi đối tượng AMF đã được nhận đầy đủ bởi client thì chỉ số này có thể được tái sử dụng.



Hình 4. *Tiêu đề RTMP 12 byte*

Đối với tiêu đề 12 byte thì 3 byte tiếp theo là trường timestamp (little-endian), 3 byte tiếp theo nữa là chiều dài của đối tượng AMF (big-endian), byte kế tiếp quy định nội dung của đối tượng AMF (xem hình ?), 4 byte cuối cùng xác định source id.

Đối với tiêu đề 8 byte thì bỏ đi trường source id trong tiêu đề 12 byte.

Đối với tiêu đề 4 byte thì bỏ đi trường length, content type trong tiêu đề 8 byte.

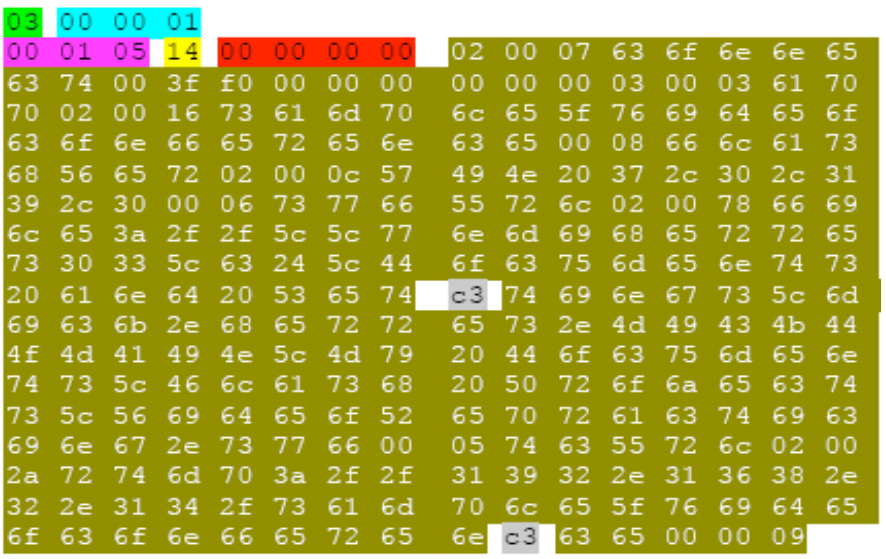
Đối với tiêu đề 1 byte thì bỏ đi trường timestamp trong tiêu đề 4 byte nghĩa là nó chỉ gồm byte đầu tiên chứa kiểu tiêu đề và chỉ số đối tượng AMF.

0×01	Chunk Size
0×02	Unknown
0×03	Bytes Read
0×04	Ping
0×05	Server BW
0×06	Client BW
0×07	Unknown
0×08	Audio Data
0×09	Video Data
0x0A - 0xE	Unknown
0x0F	Flex Stream
0×10	Flex Shared Object
0×11	Flex Message
0×12	Notify
0×13	Shared Object
0×14	Invoke

Hình 5. *Một số giá trị trong trường Content Type*

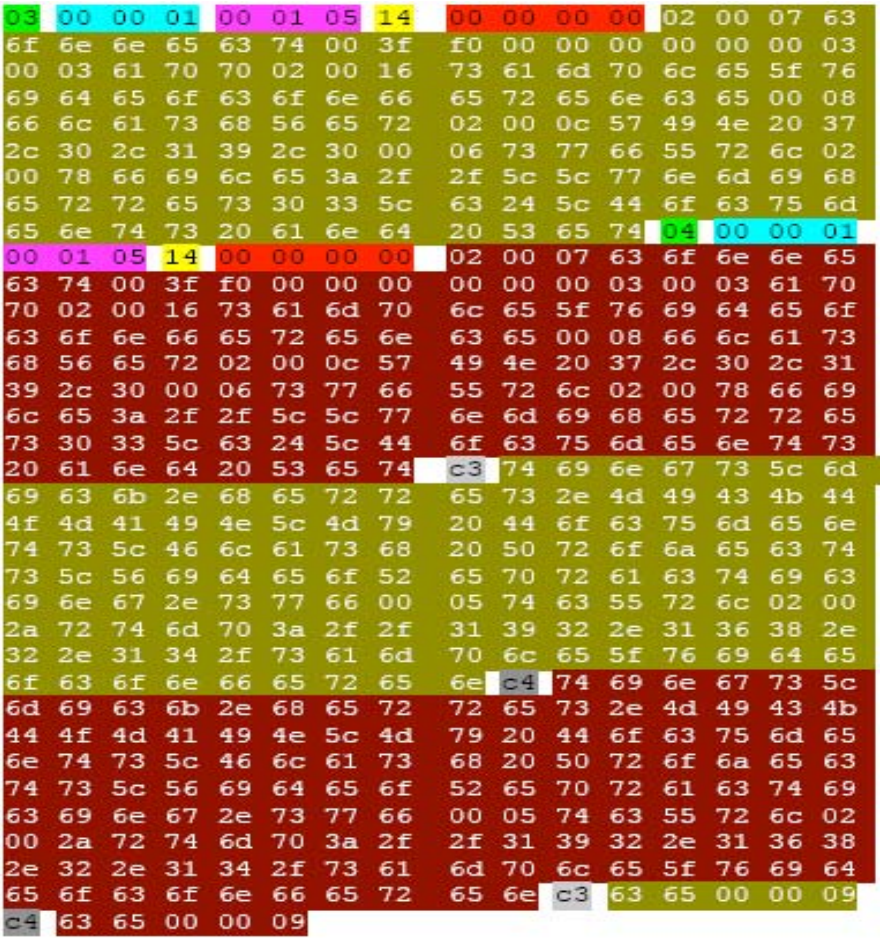
1.4. Truyền tải nhiều đối tượng AMF trên cùng một kết nối

Mỗi đối tượng AMF được chia thành các khối dữ liệu có kích thước 128 byte (ngoại trừ audio có kích thước 64 byte). Khối đầu tiên thường được gắn tiêu đề 12 byte, các khối tiếp theo thường được gắn tiêu đề 1 byte, trong bất kì loại tiêu đề nào cũng có chứa chỉ số của đối tượng AMF tương ứng.



Hình 6. *Các khối dữ liệu của một đối tượng AMF có chỉ số là 0x03*

Để có thể truyền nhiều đối tượng AMF trên một kết nối đơn người ta không truyền các khối dữ liệu của một đối tượng AMF liên tiếp nhau mà truyền xen kẽ các khối dữ liệu của nhiều đối tượng AMF.



Hình 7. *Truyền các khối dữ liệu xen kẽ nhau.*

2. Phương pháp đóng gói thông điệp AMF (ActionScript Message Format)

AMF0 là phiên bản đầu tiên của AMF giúp ghép nối các đối tượng dữ liệu, chứa thông tin về kiểu của của dữ liệu. AMF0 cũng hỗ trợ truyền các kiểu dữ liệu phức tạp bằng cách dùng tham chiếu để tránh dư thừa dữ liệu. Phiên bản tiếp theo là AMF3 được tối ưu hóa trong phương pháp biểu diễn dữ liệu.

2.1. AMF0

Trong AMF0, các kiểu dữ liệu được đánh dấu bằng một 1 byte. Theo sau đó là dữ liệu theo đúng những gì đã được mô tả ở byte đánh dấu phía trước.

Các kiểu dữ liệu của AMF0 được liệt kê ngay sau đây:

Bảng 1: *Kiểu dữ liệu trong AMF0*

Kiểu số	0x00
Kiểu logic	0x01
Kiểu chuỗi	0x02
Kiểu đối tượng	0x03
Kiểu null	0x05
Kiểu tham chiếu	0x07
Kiểu mảng ECMA	0x08
Đánh dấu kết thúc đối tượng	0x09
Kiểu mảng dày	0x0A
Kiểu ngày tháng	0x0B
Kiểu chuỗi dài	0x0C
Kiểu XML	0x0F
Kiểu lớp	0x10

- Kiểu số được dùng để mã hoá một số ActionScript. Dữ liệu theo sau một marker luôn luôn là 8 byte số dấu chấm động double IEEE-754

  Kiểu_số = đánh_dấu DOUBLE

- Kiểu logic được dùng để mã hoá kiểu nguyên thuỷ logic của ActionScript. Một byte đánh dấu là kiểu logic, theo sau đó là một byte chỉ giá trị của dữ liệu logic. Giá trị này bằng 0 là false, bằng 1 là true.

  Kiểu_logic = đánh_dấu U8 (số nguyên không dấu 8 bit)

- Kiểu chuỗi được dùng để mã hoá bất kì một chuỗi nào có độ dài nhỏ hơn 65525 kí tự. Nếu chuỗi nào dài hơn độ dài này thì sẽ chuỗi kiểu chuỗi dài.

  Kiểu_chuỗi = đánh_dấu UTF-8

- Kiểu đối tượng được dùng để mã hoá kiểu object của ActionScript. Kiểu này tương đối phức tạp được ghép nối như sau:

  Kiểu_đối_tượng = đánh_dấu (UTF-8 giá_trị)(UTF-8 giá_Trị)…(UTF-8 giá_Trị) UTF-8_rỗng đánh_dấu_kết_thúc

- Kiểu null chỉ được biểu diễn bằng byte đánh dấu kiểu, và không có thông tin đính kèm theo sau.

  Kiểu_null = đánh_dấu

- Kiểu tham chiếu được dùng để chỉ lại dối tượng đã mã hoá phía trước mà không cần phải truyền lại, tránh dư thừa dữ liệu. Kiểu dữ liệu tham chiếu sử dụng số nguyên 16 bit đề dùng như một chỉ mục và bắt đầu bằng 0.

  Kiểu_tham_chiếu = đánh dấu U16

- Kiểu mảng ECMA được xem như là một kiểu phức tạp của AMF0. Kiểu này có thể được sử dụng kèm với kiểu tham chiếu cho dữ liệu của nó. Sau byte đánh dấu là một byte chỉ kích thước của mảng, và tiếp theo sau đó là dữ liệu của mảng.

  Kiểu_mảng_ECMA = đánh_dấu U32 U32*(UTF-8 gia_tri) UTF-8_rỗng đánh_dấu_kết_thúc

- Kiểu mảng dày chứa dữ liệu cần truyền đi.

  Kiểu_mảng_dày = dánh dấu U32 U32*giá_tri

- Kiểu chuỗi dài được sử dụng trong AMF0 để mã hoá một chuỗi chiếm chiều dài nhiều 65535.

  Kiểu_chuỗi_dài = dánh_dấu UTF-8_dài

- Kiểu ngày tháng được dùng để mã hoá số lượng ms đã qua để từ ngày 1 tháng 1 năm 1970 giờ UTC

  Múi_giớ = S16 (số nguyên có dấu 16 bit)

  Kiểu_ngày_tháng = đánh_dấu DOUBLE Múi_giờ

- Kiểu XML cũng được hỗ trợ trong AMF0 khi dùng để ghép nối dữ liệu trong chuỗi XML. AMF0 coi kiểu XML giống như những kiểu chuỗi dài khác.

  Kiểu_XML = đánh_dấu UTF-8_dài

- Kiểu lớp là kiểu được dùng để ghép nối các dữ liệu trong một lớp lại thành một chuỗi gửi đi. Kiểu lớp cũng có thể được sử dụng kèm với kiểu tham chiếu.

  Tên_lớp = UTF-8

  Kiểu_lớp = đánh_dấu Tên_lớp (UTF-8 giá_trị) (UTF-8 giá_trị) … (UTF-8 giá_trị) UTF-8_rỗng đánh_dấu_kết_thúc

2.2. AMF3

AMF3 là một phiên bản mới của AMF, nhằm cải tiến những hạn chế của AMF0. Cải tiến của AMF3 là hạn chế lại việc truyền dư thừa dữ liệu và bổ sung thêm một số kiểu dữ liệu mới.

Trong AMF3, mỗi kiểu dữ liệu cũng được bắt đầu bằng một byte chiều dài, tiếp theo là dữ liệu tương ứng với dữ liệu được mô tả ở byte đánh dấu. Giá trị của byte đánh dấu có ý nghĩa như sau:

Bảng 2: *Kiểu dữ liệu của AMF3*

Kiểu undefined	0x00
Kiểu null	0x01
Kiểu logic false	0x02
Kiểu logic true	0x03
Kiểu số nguyên	0x04
Kiểu số double	0x05
Kiểu chuỗi	0x06
Kiểu văn bản XML	0x07
Kiểu ngày tháng	0x08
Kiểu mảng	0x09
Kiểu đối tượng	0x0A
Kiểu XML	0x0B
Kiểu mảng byte	0x0C

- Kiểu undefined chỉ được biểu diễn bằng byte đánh dấu không có dữ liệu đi kèm theo sau.

  Kiểu_undefined = đánh_dấu

- Kiểu null cũng giống như kiểu undefined chỉ được biểu diễn bằng byte đánh dấu và không có dữ liệu đi kèm theo sau.

  Kiểu_null = đánh_dấu

- Kiểu logic false, kiểu logic true được dùng để thay thế cho kiểu logic ở AMF0, cái giá trị logic được biểu diễn kèm theo byte đánh dấu.

  Kiểu_logic_false = đánh_dấu_logic_false

  Kiểu_logic_true = đánh_dấu_logic_true

- Kiểu số nguyên được dùng để mã hoá số nguyên không dấu 29 bit theo cú pháp ABNF (Augmented Backus-Naur Form)

  Kiểu_số_nguyên = đánh_dấu U29

- Kiểu double được dùng để mã hoá một số, kiểu này cũng được dùng thay cho kiểu số nguyên nếu số có giá trị lớn hơn 2^{29}. Kiểu này bao gồm một số dấu chấm động IEEE-754 theo sau một byte đánh dấu.

  Kiểu_double = đánh_dấu DOUBLE

- Kiểu chuỗi được dùng để biểu diễn một chuỗi trong AMF3 và dùng tương đương như chuỗi dài trong AMF0

  Kiểu_chuỗi = đánh_dấu UTF-8-vr

- Kiểu văn bản XML được dùng để mã hoá kiểu XMLDocument trong ActionScript. Chiều dài của một văn bản XML được giới hạn trong 256MB

  Kiểu_văn_bản_XML = đánh_dấu UTF-8

- Kiểu ngày tháng có ý nghĩa giống như trong AMF0.

  Kiểu_ngày_tháng = đánh_dấu (U29_1 | U29_2 DOUBLE)

  U29_1 có bit đầu tiên là 0 chỉ kiểu ngày tháng tham chiếu, U29_2 có bit đầu tiên là 1 tiếp theo sau là một DOUBLE chỉ số lượng ms từ ngày 1 tháng 1 năm 1970 tới thời điểm hiện tại tính theo giờ UTC.

- Kiểu mảng được sử dụng thay cho kiểu ECMA, mảng dày đặc của AMF0.

  Kiểu_mảng = đánh_dấu (U29_1 | U29_2 (Kiểu_ECMA | Kiểu_mảng_dày))

- Kiểu đối tượng được dùng để chất lượng đối với các kiểu Object của ActionScript và kiểu lớp của người dùng

  Kiểu_mảng = đánh_dấu (kiểu_đối_tượng | kiểu_lớp)

- Kiểu XML được dùng để mã hoá đối tượng XML của ActionScript có hỗ trợ cú pháp E4X. Nội dung của kiểu này được xử lý giống như một UTF-8

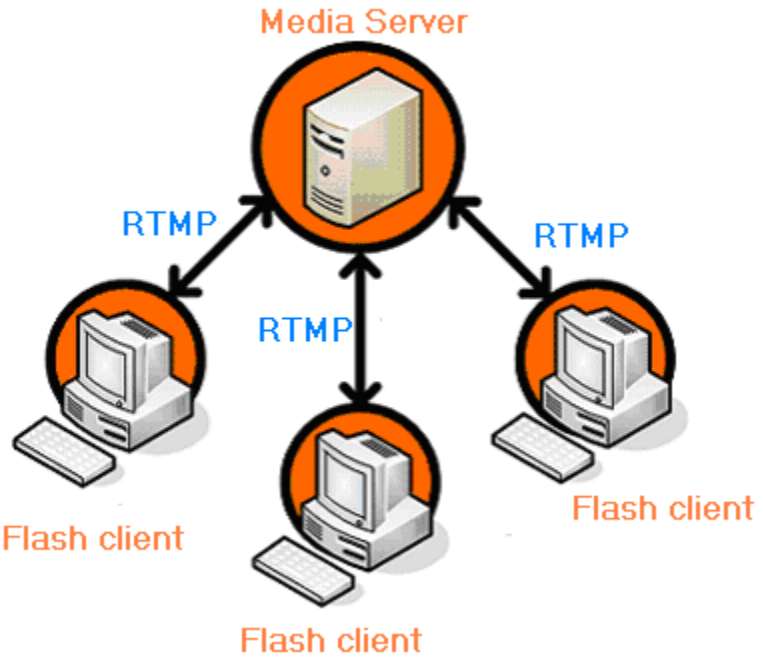
  Kiểu_XML = đánh_dấu (U29_1 | U29_2 UTF-8)

- Kiểu mảng byte là một kiểu mới được hỗ trợ trong AMF3 so với AMF0. Kểu này cũng tượng tự như kiểu mảng của AMF3 nhưng các phần tử của nó là các byte.

  Kiểu_mảng_byte = đánh_dấu (U29_1 | U29_2 n*(U8))

III. XÂY DỰNG MODULE VIDEO CONFERENCE, SHARED DESKTOP, VÀ LƯU BÀI GIẢNG

Mô hình kết nối giữa client với Media Server như sau:



Hình 8. *Mô hình kết nối giữa Flash client và Media Server*

1. Module Video Conference

1.1. Chức năng

Giúp cho giảng viên và các học viên có thể thấy được hình ảnh và nói chuyện trực tiếp với nhau.

1.2. Phương pháp thực hiện

Đầu tiên phía gửi phải tạo một kết nối tới Media Server sử dụng lớp `NetConnection`. Sau đó tạo một luồng upload bằng lớp `Netstream`, gắn đối tượng `Camera` và `Microphone` vào lớp `NetStream` và gọi hàm `publish` để gửi dữ liệu video lên Media Server.

Media Server nhận được dữ liệu từ client phía gửi sẽ tiến hành chuyển tiếp dữ liệu xuống cho client phía nhận.

Client phía nhận cũng phải tạo một kết nối tới Media Server sử dụng lớp `NetConnection` tương tự phía gửi. Sau đó tạo một luồng download bằng lớp `NetStream`, gắn luồng download vào một đối tượng hiển thị Video và gọi hàm `play` của `NetStream` để bắt đầu phát hình.

2. Module Shared Desktop

2.1. Chức năng

Nếu chỉ có Video Conference thì chưa đủ, bởi vì có những bài tập thực hành mà học viên cần theo dõi trực tiếp thao tác của giảng viên (chẳng hạn như thao tác xử lí ảnh, hay cấu hình các thiết bị …). Việc theo dõi trực tiếp thao tác của giảng viên sẽ giúp cho học viên tiếp thu nhanh hơn là làm từng bước theo tài liệu.

2.2. Phương pháp thực hiện

Hiện tại Flash chưa hỗ trợ quay phim màn hình. Nhưng có thể thực hiện việc quay phim màn hình bằng 2 phương pháp. Phương pháp thứ nhất là sử dụng thêm một Java Applet, Java Applet nay

Ở đây chúng tôi sử dụng phương pháp thứ 2. Chúng tôi sử dụng một phần mềm freeware là SplitCamera để tạo một camera giả lập có chức năng quay phim màn hình. Sau đó sử dụng lại module Video Conference ở trên để thực hiện Shared Desktop.

3. Lưu lại bài giảng

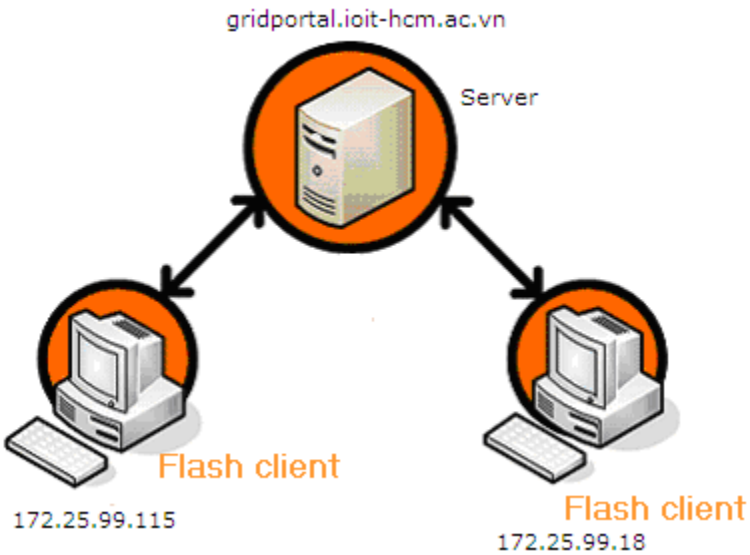
Việc lưu lại bài giảng của giảng viên rất quan trọng. Nó giúp cho những học viên không có điều kiện theo dõi trực tiếp bài giảng của giáo viên có thể xem lại bài giảng của giáo viên khi thích hợp.

Media Server chẳng hạn như Red5 có hỗ trợ chức năng lưu lại các đoạn video gửi lên từ các client. Chúng tôi sử dụng chức năng này để lưu lại bài giảng của giảng viên. Khi Flash client của giảng viên gửi dữ liệu lên Media Server chúng tôi thêm tham số record để yêu cầu Media Server lưu lại đoạn Video này.

IV. MỘT SỐ HÌNH ẢNH THỬ NGHIỆM

Flash client được viết bằng ActionScript 3, Media Server chúng tôi sử dụng là Red5 Server

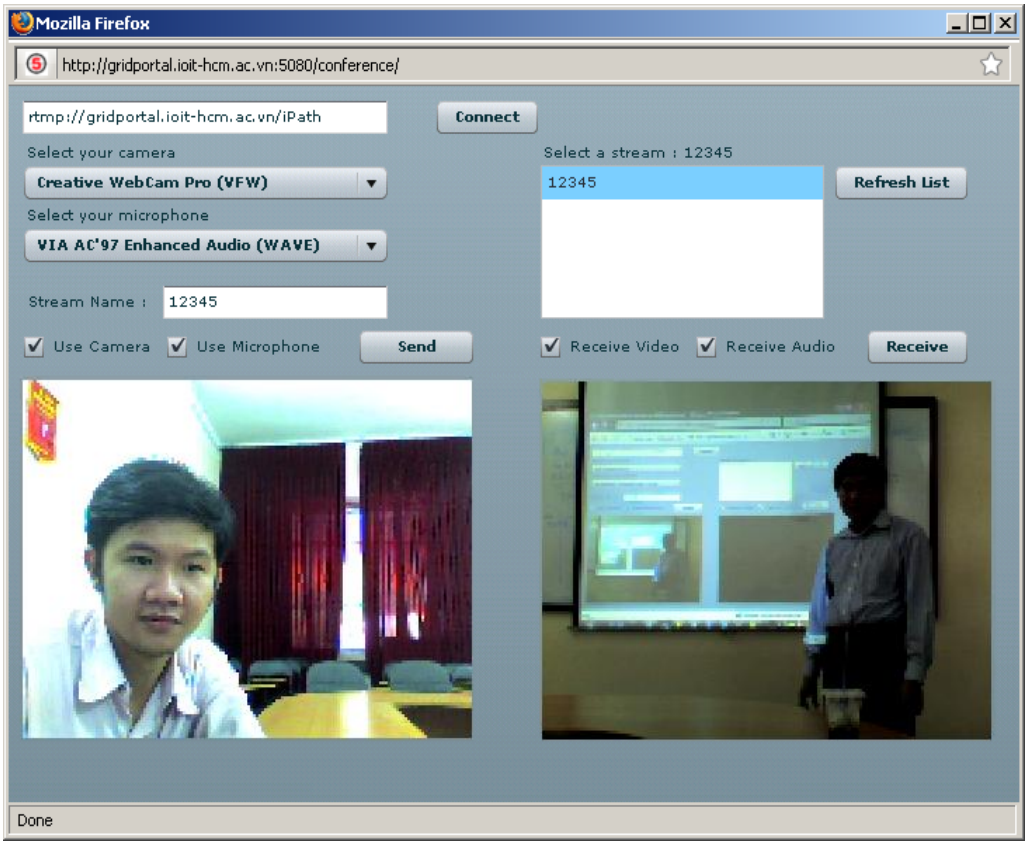
Mô hình thử nghiệm:



Hình 9. *Mô hình thử nghiệm*

Thử nghiệm Video Conference

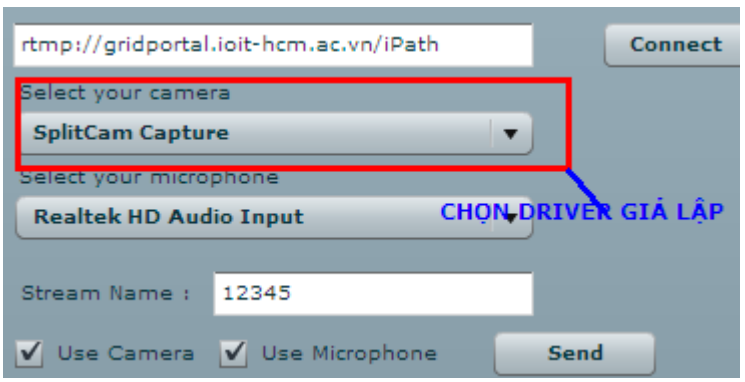
Máy 172.25.99.18 ghi hình giảng viên đang giảng bài và chuyển phát cho học viên trên máy 172.25.99.115 xem.



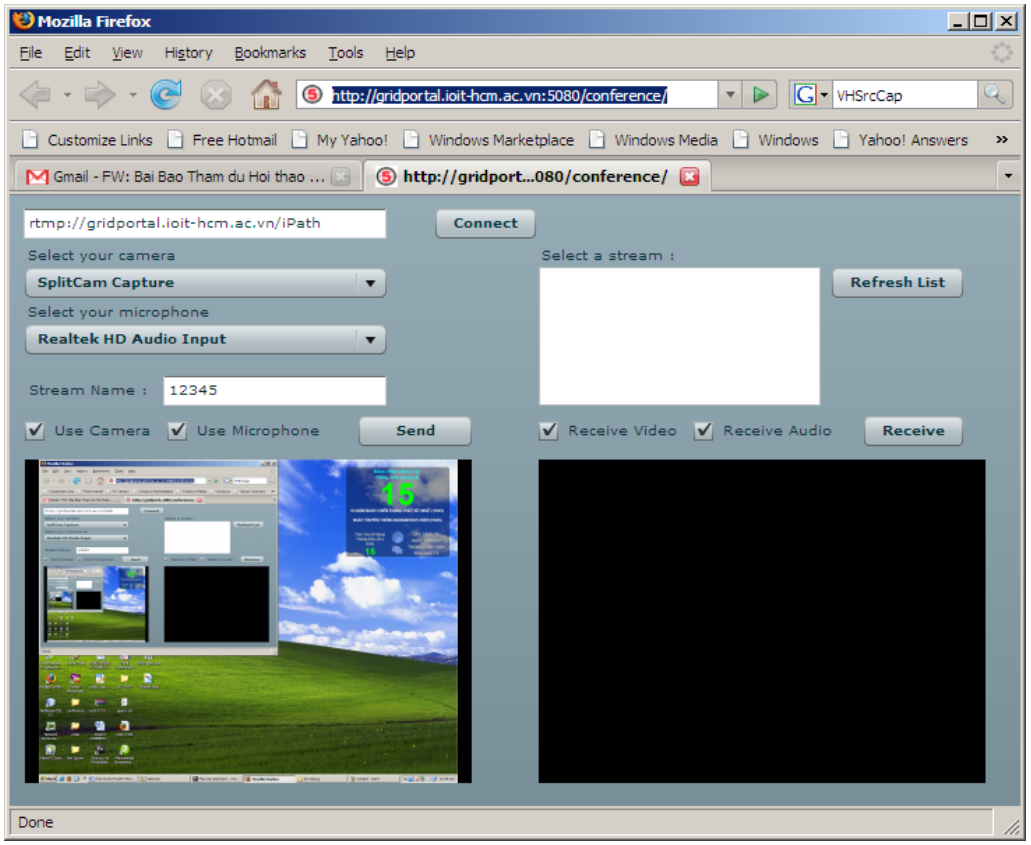
Hình 10. *Thử nghiệm module Video Conference*

Thử nghiệm Shared Desktop

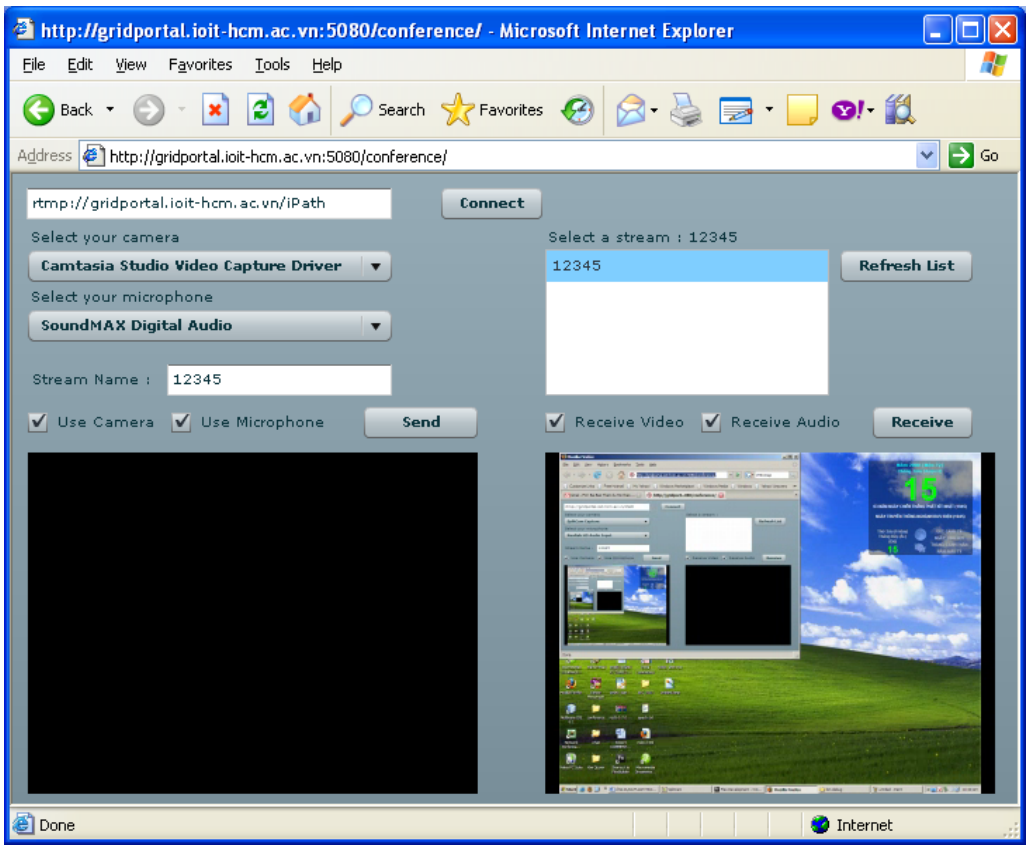
Màn hình Desktop trên máy 172.25.99.115 được truyền qua cho máy 172.25.99.18



Hình 11. *Chọn thiết bị Camera giả lập để quay phim màn hình*



Hình 12. *Màn hình Desktop trên máy 172.25.99.115 được ghi lại*



Hình 13. *Màn hình Desktop trên máy 172.25.99.118 được nhìn từ máy 172.25.99.18*

V. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Chúng tôi đã thực hiện một số tính năng hỗ trợ E-learning như Video Conference, Share Desktop, lưu bài giảng và đã có một số kết quả như trình bày ở phần trên. Tuy nhiên đó còn là những module ở giai đoạn thử nghiệm. Để có thể sử dụng được trong E-learning thì cần gắn kết các module đó lại với nhau, đồng thời chỉnh sửa lại giao diện làm cho người sử dụng cảm thấy tiện lợi hơn. Ngoài ra cần hỗ trợ thêm một số tính năng khác như WhiteBoard, Chat, …

Chương trình của chúng tôi sử dụng Flash với chuẩn nén video H.264 có hệ số nén cao, đồng thời hỗ trợ hàm điều chỉnh chất lượng video trong một khoảng khá rộng nên có thể sử dụng trên nhiều đường truyền có tốc độ khác nhau.

TÀI LIỆU THAM KHảO

[1]. Gnash, http://wiki.gnashdev.org/RTMP.

[2]. Red5, http://osflash.org/red5.

[3]. http://livedoc.adobe.com.