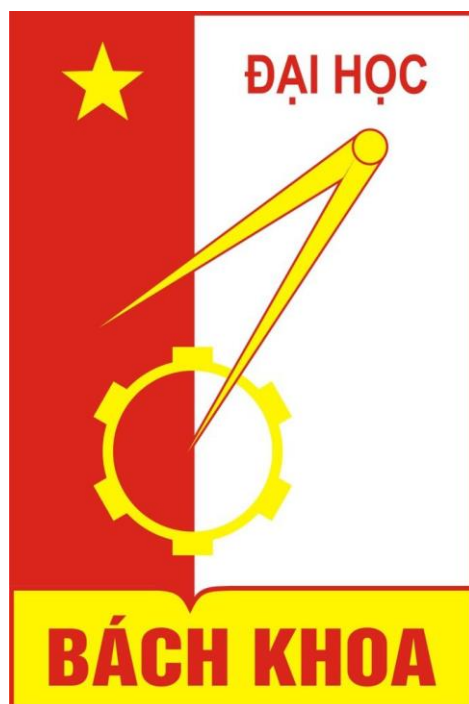


**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**  
**VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

\*\*\*



**BÀI TẬP LỚN MẠNG MÁY TÍNH**  
**TÌM HIỂU GIAO THỨC FILE TRANSFER PROTOCOL**

**GIẢNG VIÊN HƯỚNG DẪN: PGS.TS NGÔ QUỲNH THU**

**HỌ TÊN SINH VIÊN: ĐÀM MINH TIẾN                      20156599**

**NGUYỄN VĂN TRỌNG                      2015**

**NGUYỄN TRỌNG TÀI                      2015**

**NGUYỄN VĂN THẮNG                      2015**



# MỤC LỤC

<b>PHẦN I: GIỚI THIỆU .....</b>	<b>3</b>
<b>PHẦN II: LỊCH SỬ PHÁT TRIỂN .....</b>	<b>4</b>
1. Giai đoạn 1971 – 1980.....	4
2. Giai đoạn tiếp theo đến năm 1997 .....	4
3. Giai đoạn 1997 đến nay .....	4
<b>PHẦN III: CÁCH THỨC HOẠT ĐỘNG.....</b>	<b>6</b>
1. Mô hình hoạt động của FTP, các thành phần trong giao thức, và các thuật ngữ cơ bản.....	6
2. Thiết lập kênh điều khiển và chứng thực người dùng trong FTP:.....	9
3. Quản lý kênh dữ liệu FTP, kết nối kênh dữ liệu dạng chủ động và bị động.....	10
4. Các phương thức truyền dữ liệu trong FTP .....	15
<b>PHẦN IV: PHÂN LOẠI.....</b>	<b>17</b>
1. SFTP.....	17
2. FTPS.....	18
3. SFTP.....	18
4. TFPT .....	19
<b>PHẦN V: MÔ PHỎNG GIAO THỨC .....</b>	<b>21</b>
1. Phần mềm mô phỏng:.....	21
2. Cài đặt và mô phỏng:.....	21
2.1 Khởi tạo các đối tượng: .....	21
2.2. Thiết lập các kết nối:.....	22
2.3. Thiết lập thuộc tính các đối tượng: .....	22
2.4. Test kết nối tới FTPServer trên cmd với lệnh ping .....	25
2.5. Tải tệp tin lên / từ FTPServer: .....	27
<b>PHẦN VI: PHÂN TÍCH VÀ KẾT LUẬN .....</b>	<b>29</b>
1. Phân tích ưu nhược điểm của FTP: .....	29
2. Kết luận: .....	30

## PHẦN I: GIỚI THIỆU

*File Transfer Protocol là một trong những giao thức truyền tệp trên mạng quan trọng, cung cấp các phương tiện để các máy tính có thể chia sẻ những tệp tin với nhau. Kể từ khi cơ chế truyền tệp tin được đề cập lần đầu tiên trong bản phác thảo ở MIT – RFC 114 (1971), File Transfer Protocol chính thức ra đời năm 1973 trong bản thảo RFC 454, File Transfer Protocol luôn được nghiên cứu và phát triển các bản thảo sau này 686, 765, 959, 1579, 2228,... như một sự chứng tỏ cho tầm quan trọng của giao thức này.*

*Báo cáo này với mục đích tìm hiểu khái quát về giao thức truyền tệp trên mạng File Transfer Protocol cũng như có thể mô phỏng được cách thức hoạt động trên phần mềm Cisco Packet Tracer.*

Nhóm sinh viên

Đàm Minh Tiến

Nguyễn Văn Thắng

Nguyễn Văn Trọng

Nguyễn Trọng Tài

## PHẦN II: LỊCH SỬ PHÁT TRIỂN

Giao thức FTP được phát triển bởi Abhay Bhushan, chuẩn đầu tiên là RFC 114 xuất bản trong tháng 4 năm 1971, và chính thức ra đời năm 1973. Tính đến hiện nay FTP phát triển theo 3 giai đoạn. Dưới đây là tóm tắt 1 vài chuẩn RFC trong từng khoảng thời gian phát triển của FTP:

### 1. Giai đoạn 1971 – 1980

a. Trong khoảng thời gian 1971- 1980 ,là sự ra đời cũng là khoảng thời gian liên tiếp phát hành các chuẩn RFC 172, RFC 265, RFC 354 và những lệnh cơ bản của FTP được định nghĩa.

b. Vào năm 1980 chuẩn FTP RFC 765 ra đời bởi Jon Postel của ITI.

c. TFTP được chuẩn hóa vào năm 1981 là một giao thức FTP đơn giản cho phép khác hàng lấy tệp lên máy chủ từ xa và có thể tìm thấy trong RFC 1350.

### 2. Giai đoạn tiếp theo đến năm 1997

a. Năm 1985 chuẩn FTP RFC 959 ra đời bởi Joyce Reynolds của ISI.

b. Tháng 2 năm 1994 chuẩn RFC 1579 được ra đời với mục đích cho phép FTP thân thiện và hoạt động với tường lửa.

### 3. Giai đoạn 1997 đến nay

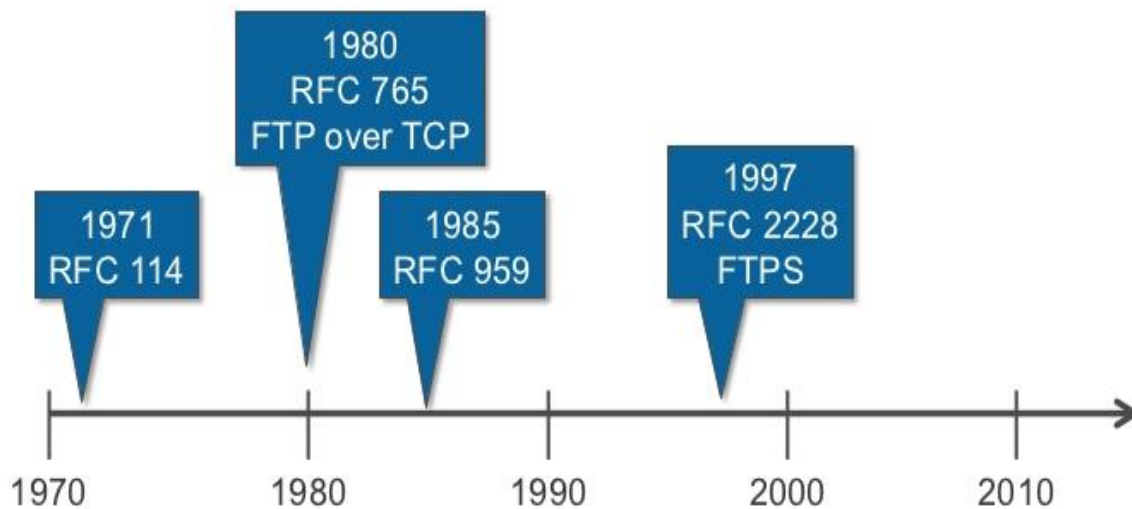
a. Năm 1997 là sự ra đời giao thức FTPS áp dụng cho FTP để tăng tính bảo mật, và đó là sự ra đời của chuẩn RFC 2228 và chuẩn RFC 4217. FTPS ( còn được gọi là FTPES, FTPS-SSL, S-FTP và FTP secure) là một phần mở rộng cho FTP và được sử dụng phổ biến mà hỗ trợ cho Secure Sockets Layer(SSL) là những giao thức mật mã.

b. Tháng 9 năm 1998, để hỗ trợ hoạt động giao thức Ipv6, chuẩn RFC 2428 được phát hành.

c. Vào những năm tiếp theo là sự ra đời của SSH File transfer protocol ( Secure File Transfer Protocol, hay SFTP ), nó được thiết kế bởi IETF với tính bảo mật rất cao và đó cũng là thời điểm ra đời của chuẩn RFC 4251.

-----Dưới đây là tóm tắt về sự ra đời của 1 số chuẩn trong những khoảng thời gian khác nhau.

## FTP Over the Years



## PHẦN III: CÁCH THỨC HOẠT ĐỘNG

### 1. Mô hình hoạt động của FTP, các thành phần trong giao thức, và các thuật ngữ cơ bản

Giao thức FTP được mô tả một cách đơn giản thông qua mô hình hoạt động của FTP. Mô hình này chỉ ra các nguyên tắc mà một thiết bị phải tuân theo khi tham gia vào quá trình trao đổi file, cũng như về hai kênh thông tin cần phải thiết lập giữa các thiết bị đó. Nó cũng mô tả các thành phần của FTP được dùng để quản lý các kênh này ở cả hai phía – truyền và nhận. Do đó, mô hình này tạo cho ta một khởi điểm lý tưởng để xem xét hoạt động của FTP ở mức khái quát.

#### *Tiến trình Server-FTP và User-FTP*

FTP là một giao thức dạng client/server truyền thống, tuy nhiên thuật ngữ client thông thường được thay thế bằng thuật ngữ user – người dùng – do thực tế là người sử dụng mới là đối tượng trực tiếp thao tác các lệnh FTP trên máy clients. Bộ phần mềm FTP được cài đặt trên một thiết bị được gọi là một tiến trình. Phần mềm FTP được cài đặt trên máy Server được gọi là tiến trình Server-FTP, và phần trên máy client được gọi là tiến trình User-FTP.

#### *Kênh điều khiển và kênh dữ liệu trong FTP*

Một khái niệm cốt yếu mà ta cần phải nắm về FTP là: mặc dù giao thức này sử dụng kết nối TCP, nhưng nó không chỉ dùng một kênh TCP như phần lớn các giao thức truyền thông khác. Mô hình FTP chia quá trình truyền thông giữa bộ phận Server với bộ phận client ra làm hai kênh logic:

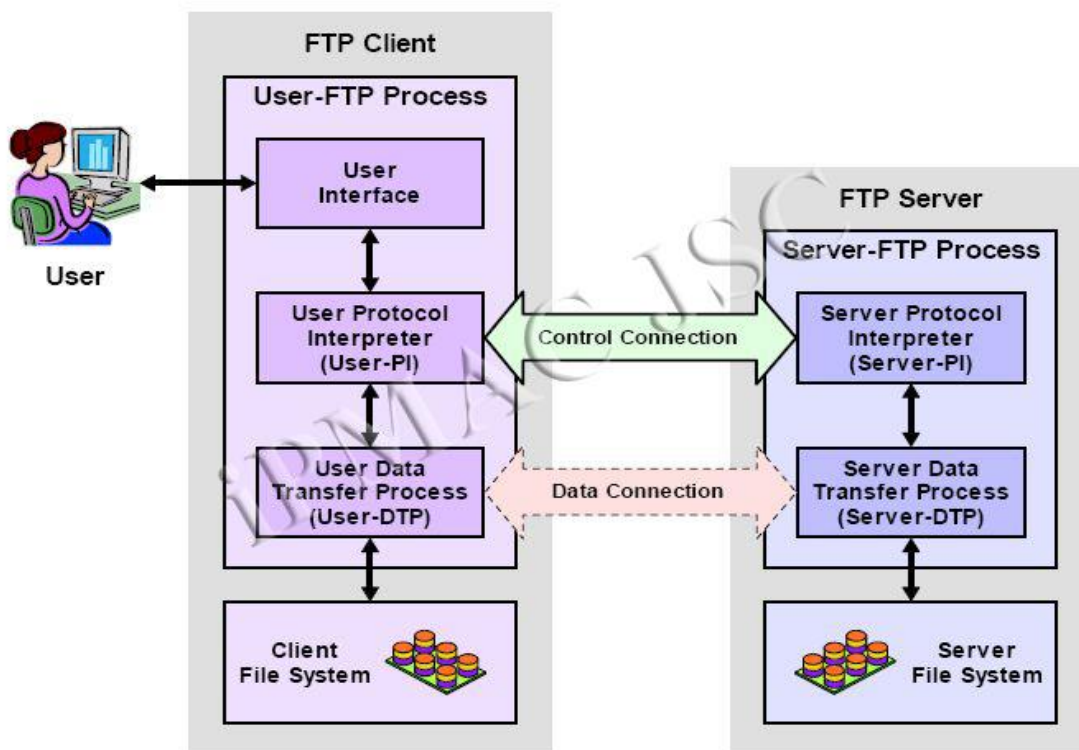
- Kênh điều khiển: đây là kênh logic TCP được dùng để khởi tạo một phiên kết nối FTP. Nó được duy trì xuyên suốt phiên kết nối FTP và được sử dụng chỉ để truyền các thông tin điều khiển, như các lệnh và các hồi đáp trong FTP. Nó không được dùng để truyền file
- Kênh dữ liệu: Mỗi khi dữ liệu được truyền từ server tới client, một kênh kết nối TCP nhất định lại được khởi tạo giữa chúng. Dữ liệu được truyền đi qua kênh kết nối này – do đó nó được gọi là kênh dữ liệu. Khi file được truyền

xong, kênh này được ngắt. Việc sử dụng các kênh riêng lẻ như vậy tạo ra sự linh hoạt trong việc truyền truyền dữ liệu – mà ta sẽ thấy trong các phần tiếp theo. Tuy nhiên, nó cũng tạo cho FTP độ phức tạp nhất định.

### *Các tiến trình và thuật ngữ trong FTP*

Do các chức năng điều khiển và dữ liệu sử dụng các kênh khác nhau, nên mô hình hoạt động của FTP cũng chia phần mềm trên mỗi thiết bị ra làm hai thành phần logic tương ứng với mỗi kênh. Thành phần Protocol Interpreter (PI) là thành phần quản lý kênh điều khiển, với chức năng phát và nhận lệnh. Thành phần Data Transfer Process (DTP) có chức năng gửi và nhận dữ liệu giữa phía client với server. Ngoài ra, cung cấp cho tiến trình bên phía người dùng còn có thêm thành phần thứ ba là giao diện người dùng FTP - thành phần này không có ở phía server.

Do đó, có hai tiến trình xảy ra ở phía server, và ba tiến trình ở phía client. Các tiến trình này được gắn với mô hình FTP để mô tả chi tiết hoạt động của giao thức FTP. Dưới đây là hình đối chiếu các tiến trình vào trong mô hình FTP:





*Các tiến trình phía server:*

Các tiến trình phía server bao gồm hai giao thức:

- Server Protocol Interpreter (Server-PI): chịu trách nhiệm quản lý kênh điều khiển trên server. Nó lắng nghe yêu cầu kết nối hướng tới từ users trên cổng dành riêng. Khi kết nối đã được thiết lập, nó sẽ nhận lệnh từ phía User-PI, trả lời lại, và quản lý tiến trình truyền dữ liệu trên server.
- Server DataTransfer Process (Server-DTP): làm nhiệm vụ gửi hoặc nhận file từ bộ phận User-DTP. Server-DTP vừa làm nhiệm vụ thiết lập kết nối kênh dữ liệu và lắng nghe một kết nối kênh dữ liệu từ user. Nó tương tác với server file trên hệ thống cục bộ để đọc và chép file.

*Các tiến trình phía client:*

Các tiến trình phía client bao gồm hai giao thức:

- User Protocol Interpreter (User-PI): chịu trách nhiệm quản lý kênh điều khiển phía client. Nó khởi tạo phiên kết nối FTP bằng việc phát ra yêu cầu tới phía Server-PI. Khi kết nối đã được thiết lập, nó xử lý các lệnh nhận được trên giao diện người dùng, gửi chúng tới Server-PI, và nhận phản hồi trở lại. Nó cũng quản lý tiến trình User-DTP.
- User Data Transfer Process (User-DTP): là bộ phận DTP nằm ở phía người dùng, làm nhiệm vụ gửi hoặc nhận dữ liệu từ Server-DTP. User-DTP có thể thiết lập hoặc lắng nghe yêu cầu kết nối kênh dữ liệu trên server. Nó tương tác với thiết bị lưu trữ file phía client.

User Interface, cung cấp giao diện xử lý cho người dùng. Nó cho phép sử dụng các lệnh đơn giản hướng người dùng, và cho phép người điều khiển phiên FTP theo dõi được các thông tin và kết quả xảy ra trong tiến trình.

## 2. Thiết lập kênh điều khiển và chứng thực người dùng trong FTP:

Mô hình hoạt động của FTP mô tả rõ các kênh dữ liệu và điều khiển được thiết lập giữa FTP client và FTP server. Trước khi kết nối được sử dụng để thực sự truyền file, kênh điều khiển cần phải được thiết lập. Một tiến trình chỉ định sau đó được dùng để tạo kết nối và tạo ra phiên FTP lâu bền giữa các thiết bị để truyền files.

Như trong các giao thức client/server khác, FTP server tuân theo một luật passive trong kênh điều khiển. Bộ phận Server Protocol Interpreter (Server-PI) sẽ lắng nghe cổng TCP dành riêng cho kết nối FTP là cổng 21. Phía User-PI sẽ tạo kết nối bằng việc mở một kết nối TCP từ thiết bị người dùng tới server trên cổng đó. Nó sử dụng một cổng bất kỳ làm cổng nguồn trong phiên kết nối TCP.

Khi TCP đã được cài đặt xong, kênh điều khiển giữa các thiết bị sẽ được thiết lập, cho phép các lệnh được truyền từ User-PI tới Server-PI, và Server-PI sẽ đáp trả kết quả là các mã thông báo. Bước đầu tiên sau khi kênh đã đi vào hoạt động là bước đăng nhập của người dùng (login sequence). Bước này có hai mục đích:

- *Access Control* - Điều khiển truy cập: quá trình chứng thực cho phép hạn chế truy cập tới server với những người dùng nhất định. Nó cũng cho phép server điều khiển loại truy cập như thế nào đối với từng người dùng.
- *Resource Selection* - Chọn nguồn cung cấp: Bằng việc nhận dạng người dùng tạo kết nối, FTP server có thể đưa ra quyết định sẽ cung cấp những nguồn nào cho người dùng đã được nhận dạng đó.

### *Trình tự truy cập và chứng thực FTP*

Quy luật chứng thực trong FTP khá đơn giản, chỉ là cung cấp username/password.

Trình tự của việc chứng thực như sau:

1. Người dùng gửi một username từ User-PI tới Server-PI bằng lệnh USER. Sau đó password của người dùng được gửi đi bằng lệnh PASS.
2. Server kiểm tra tên người dùng và password trong database người dùng của nó. Nếu người dùng hợp lệ, server sẽ gửi trả một thông báo tới người dùng rằng phiên kết nối đã đ. c mở. Nếu người dùng không hợp lệ, server yêu cầu

người dùng thực hiện lại việc chứng thực. Sau một số lần chứng thực sai nhất định, server sẽ ngắt kết nối.

Giả sử quá trình chứng thực đã thành công, server sau đó sẽ thiết lập kết nối để cho phép từng loại truy cập đối với người dùng được cấp quyền. Một số người dùng chỉ có thể truy cập vào một số file nhất định, hoặc vào một số loại file nhất định. Một số server có thể cấp quyền cho một số người dùng đọc và viết lên server, trong khi chỉ cho phép đọc đối với những người dùng khác. Người quản trị mạng có thể nhờ đó mà đáp ứng đúng các nhu cầu truy cập FTP.

Một khi kết nối đã được thiết lập, server có thể thực hiện các lựa chọn tài nguyên dựa vào nhận diện người dùng. Ví dụ: trên một hệ thống nhiều người dùng, người quản trị có thể thiết lập FTP để khi có bất cứ người dùng nào kết nối tới, anh ta sẽ tự động được đưa tới "home directory" của chính anh ta. Lệnh tùy chọn ACCT (account) cũng cho phép người dùng chọn một tài khoản cá nhân nào đó nếu như anh ta có nhiều hơn một tài khoản.

### *Mở rộng về bảo mật FTP*

Giống như phần lớn các giao thức cũ, phương pháp đăng nhập đơn giản của FTP là một sự kế thừa từ những giao thức ở thời kỳ đầu của Internet. Ngày nay, nó không còn bảo đảm tính an toàn cần thiết trên môi trường Internet toàn cầu vì username và password được gửi qua kênh kết nối điều khiển dưới dạng clear text. Điều này làm cho các thông tin đăng nhập có thể bị nghe lén. Chuẩn RFC 2228 về các phần mở rộng cho bảo mật FTP đã định ra thêm nhiều tùy chọn chứng thực và mã hóa phức tạp cho những ai muốn tăng thêm mức độ an toàn vào trong phần mềm FTP của họ.

### **3. Quản lý kênh dữ liệu FTP, kết nối kênh dữ liệu dạng chủ động và bị động**

Kênh điều khiển được tạo ra giữa Server-PI và User-PI sử dụng quá trình thiết lập kết nối và chứng thực được duy trì trong suốt phiên kết nối FTP. Các lệnh và các hồi đáp được trao đổi giữa bộ phận PI (Protocol Interpreter) qua kênh điều khiển, nhưng dữ liệu thì không.

Mỗi khi cần phải truyền dữ liệu giữa server và client, một kênh dữ liệu cần phải được tạo ra. Kênh dữ liệu kết nối bộ phận User-DTP với Server-DTP. Kết nối này cần thiết cho cả hoạt động chuyển file trực tiếp (gửi hoặc nhận một file) cũng như đối với việc truyền dữ liệu ngầm, như là yêu cầu một danh sách file trong thư mục nào đó trên server.

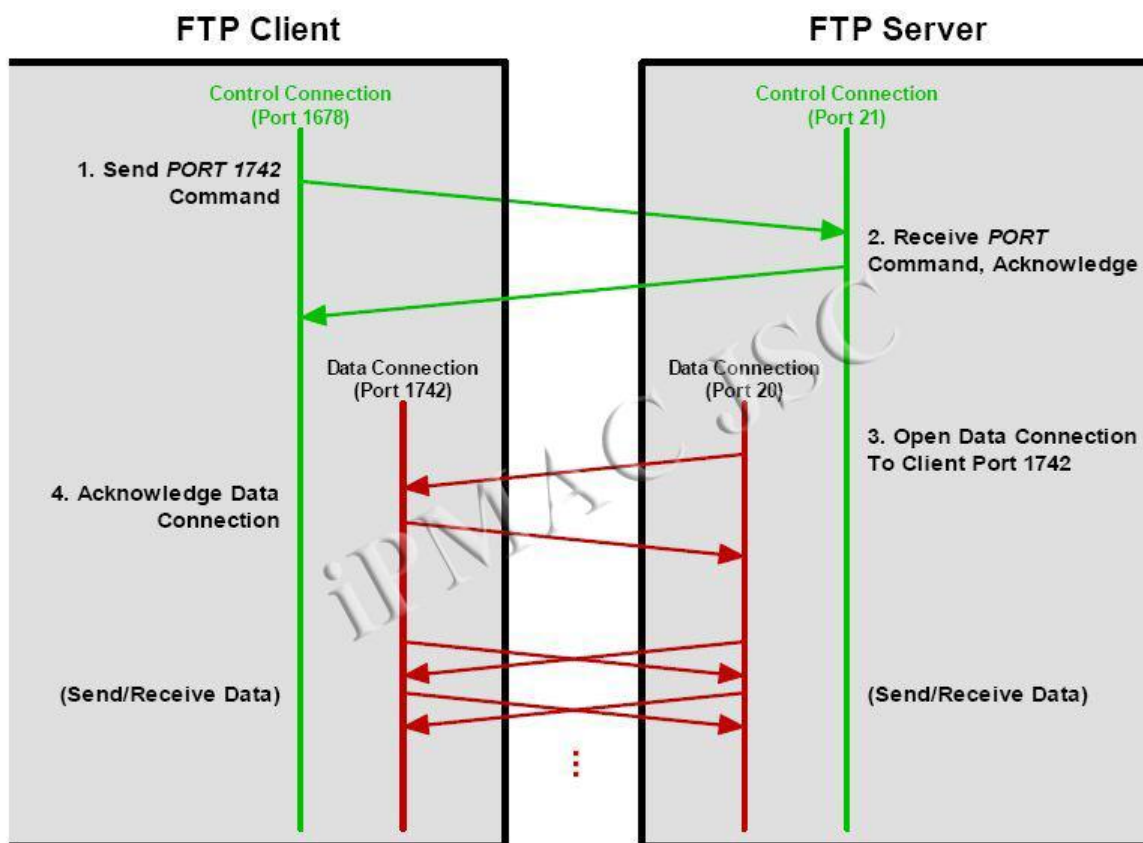
Chuẩn FTP chỉ định hai phương thức khác nhau để tạo ra kênh dữ liệu. Khác biệt chính của hai phương thức đó là ở mặt thiết bị: phía client hay phía server là phía đã đưa ra yêu cầu khởi tạo kết nối. Điều này nghe qua có vẻ khá đơn giản, nhưng kỳ thực nó lại khá quan trọng.

### *Kết nối kênh dữ liệu dạng chủ động*

Phương thức đầu tiên đôi khi còn được gọi là kết nối kênh dữ liệu dạng thông thường (vì nó là phương pháp mặc định) và đôi khi được gọi là kết nối dạng chủ động (để đối chiếu với dạng kết nối bị động mà ta sẽ xét ở phần sau). Trong dạng kết nối này, phía Server-DTP khởi tạo kênh dữ liệu bằng việc mở một cổng TCP cho phía User-DTP. Phía server sử dụng cổng được dành riêng, là cổng 20 cho kênh dữ liệu. Trên máy client, một giá trị cổng được chọn theo mặc định chính là cổng được sử dụng đối với kênh điều khiển, tuy nhiên phía client sẽ luôn chọn hai cổng riêng biệt cho hai kênh này.

Giả sử phía User-PI thiết lập một kết nối điều khiển từ cổng bất kỳ của nó là 1678 tới cổng điều khiển trên server là cổng 21. Khi đó, để tạo một kênh dữ liệu cho việc truyền dữ liệu, phía Server-PI sẽ báo cho phía Server-DTP khởi tạo một kênh kết nối TCP từ cổng 20 tới cổng 1678 của phía client. Sau khi phía client chấp nhận kênh được khởi tạo, dữ liệu sẽ được truyền đi.

Thực tế, việc sử dụng cùng một cổng cho cả kênh dữ liệu và kênh điều khiển không phải là một ý hay, nó làm cho hoạt động của FTP trở nên phức tạp. Do đó, phía client nên chỉ định sử dụng một cổng khác bằng việc sử dụng lệnh PORT trước khi truyền dữ liệu. Ví dụ: giả sử phía client chỉ định cổng 1742 với lệnh PORT. Phía Server-DTP sau đó sẽ tạo ra một kết nối từ cổng 20 của nó tới cổng 1742 phía client thay vì cổng 1678 như mặc định. Quá trình này được mô tả trong hình dưới đây.



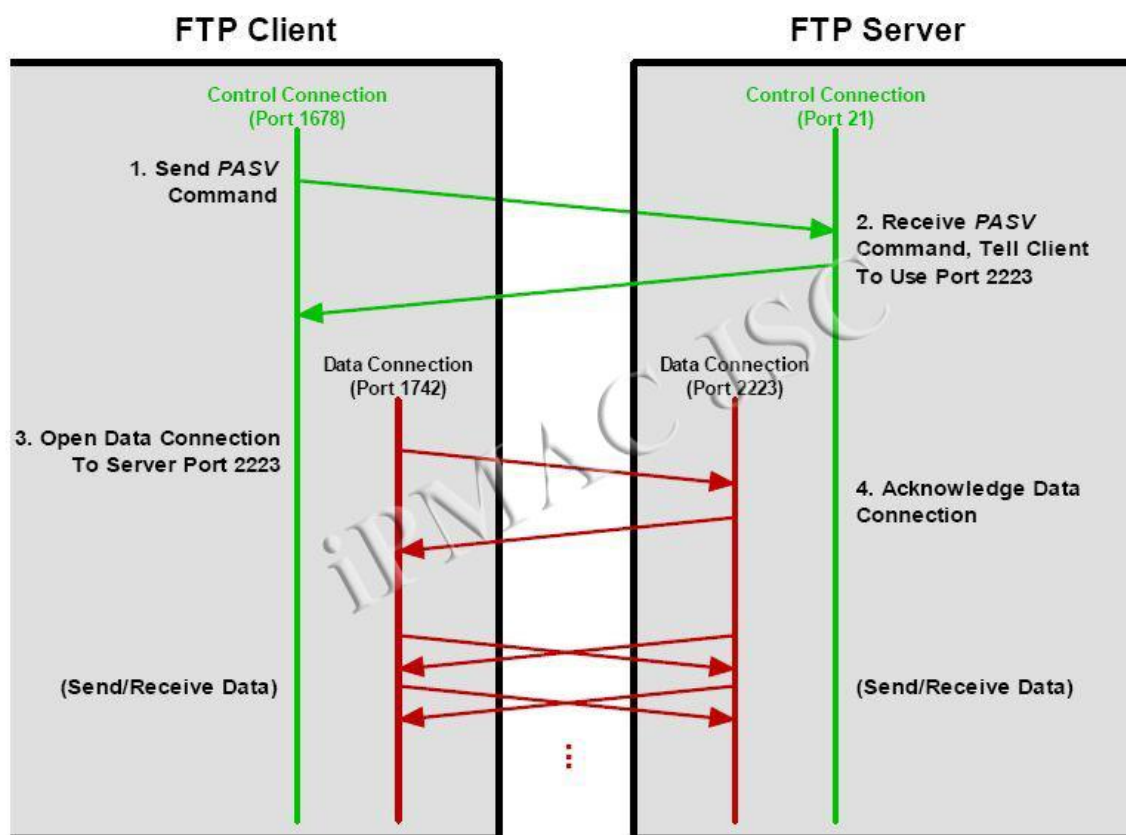
Thông thường, đối với kênh dữ liệu FTP, phía server sẽ khởi tạo việc truyền dữ liệu bằng cách mở kết nối dữ liệu tới client.

Trong trường hợp trên, phía client trước tiên sẽ đưa ra lệnh PORT để yêu cầu server sử dụng cổng 1742. Sau đó, server sẽ mở kết nối kênh dữ liệu từ cổng 20 mặc định của nó tới cổng 1742 phía client. Dữ liệu sau đó sẽ được truyền giữa các thiết bị qua các cổng này.

### *Kết nối kênh dữ liệu dạng bị động*

Phương pháp kế tiếp được gọi là kết nối dữ liệu dạng bị động. Phía client sẽ nhận server là phía bị động, làm nhiệm vụ chấp nhận một yêu cầu kết nối kênh dữ liệu được khởi tạo từ phía client. Server trả lời lại phía client với địa chỉ IP cũng như địa chỉ cổng mà nó sẽ sử dụng. Phía Server-DTP sau đó sẽ lắng nghe một kết nối TCP từ phía User-DTP trên cổng này.

Mặc định, phía client sử dụng cùng một cổng đối với cả hai kênh điều khiển và dữ liệu như trong trường hợp kết nối chủ động ở trên. Tuy nhiên, ở đây, một lần nữa phía client có thể chọn sử dụng một giá trị cổng khác cho kênh dữ liệu. Ta sẽ xét lại ví dụ ở trên một lần nữa, với cổng điều khiển phía client là 1678 tới cổng 21 phía server. Nhưng lần này truyền dữ liệu theo phương thức kết nối bị động, như mô tả trong hình dưới đây:



Phía client sẽ sử dụng lệnh PASV để yêu cầu server rằng nó muốn dùng phương thức điều khiển dữ liệu bị động. Phía Server-PI sẽ trả lời lại phía client với một giá trị cổng mà client sẽ sử dụng, từ cổng 2223 trên nó. Sau đó phía Server PI sẽ hướng cho phía Server-DTP lắng nghe trên cổng 2223. Phía User-PI cũng sẽ hướng cho



phía User-DTP tạo một phiên kết nối từ cổng 1742 phía client tới cổng 2223 phía server. Sau khi Server chấp nhận kết nối này, dữ liệu bắt đầu được truyền đi.

*Các vấn đề về tính hiệu quả và tính bảo mật trong việc chọn một phương thức kết nối*

Vấn đề phía nào là phía khởi tạo kết nối kênh dữ liệu đưa ra một câu hỏi: sự khác nhau giữa hai phương thức là gì? Điều này cũng giống như việc hỏi ai đã thực hiện một cuộc điện thoại nội bộ. Câu trả lời là sự bảo mật. Việc FTP sử dụng nhiều hơn một kết nối TCP có thể giải quyết các vấn đề về phần mềm cũng như về phần cứng mà người dùng cần phải có để đảm bảo sự an toàn cho hệ thống của họ.

*Khi xem xét việc gì sẽ xảy ra trong trường hợp kênh dữ liệu chủ động như trong ví dụ phía trên:*

Đối với phía client, có một kênh kết nối điều khiển được thiết lập từ cổng 1678 client tới cổng 21 server. Nhưng kênh dữ liệu lại được khởi tạo từ phía server. Do đó, client sẽ nhận được một yêu cầu kết nối tới cổng 1678 (hoặc cổng nào khác). Một số client sẽ nghi ngờ về việc nhận được những kết nối tới như vậy, vì trong tình huống thông thường, client mới là phía khởi tạo kết nối chứ không phải đáp trả kết nối. Do các kênh kết nối TCP hướng tới có thể mang theo những mối đe dọa nhất định, một số client có thể sẽ ngăn chặn các luồng kết nối hướng tới bằng việc sử dụng tường lửa.

Tại sao người ta lại không làm cho phía client luôn chấp nhận kết nối từ một chỉ số port được dùng trong kênh điều khiển? Vấn đề ở đây là vì client thường dùng các cổng khác nhau cho mỗi phiên kết nối bằng việc sử dụng câu lệnh PORT. Và tại sao điều này lại được thực hiện? Vì theo luật TCP: sau khi một kết nối được đóng lại, có một khoảng thời gian trống trước khi cổng đó có thể được sử dụng lại – điều này để ngăn ngừa tình trạng các phiên kết nối liên tiếp bị lẫn với nhau. Điều này sẽ tạo ra độ trễ khi gửi nhiều file – do đó phía client thường dùng các giá trị cổng khác nhau cho mỗi kết nối. Điều này rất hiệu quả nhưng cũng dẫn tới việc firewall của client sẽ hỏi có chấp nhận phiên kết nối tới với nhiều giá trị cổng không ổn định hay không.

Việc dùng kết nối kiểu kênh gián tiếp sẽ giảm thiểu vấn đề này một cách hiệu quả. Phần lớn các tường lửa có nhiều vấn đề liên quan tới kết nối hướng về với các giá trị cổng bất kỳ, hơn là gặp vấn đề với các kết nối hướng đi. Ta có thể xem chi tiết hơn về vấn đề này trong chuẩn RFC 1579. Chuẩn này khuyến nghị rằng phía client nên sử dụng kết nối kiểu bị động làm dạng mặc định thay vì sử dụng kiểu kết nối dạng chủ động cùng với lệnh PORT, để ngăn chặn tình trạng block theo cổng.

Tất nhiên, phương thức kết nối kiểu bị động không hoàn toàn giải quyết được vấn đề, chúng chỉ đẩy vấn đề về phía server mà thôi. Phía server, giờ đây phải đối mặt với việc có nhiều kênh kết nối hướng về trên hàng loạt các cổng khác nhau. Tuy nhiên việc xử lý các vấn đề bảo mật trên một nhóm nhỏ server vẫn dễ hơn nhiều so với việc phải đối mặt với một lượng lớn các vấn đề từ nhiều client. FTP server phải được cấu hình chấp nhận phương thức truyền bị động từ client, do đó cách thông thường để thiết lập trên server là thiết lập chấp nhận một số cổng kết nối hướng về trên server trong khi vẫn khóa các yêu cầu kết nối hướng về trên các cổng khác.

#### 4. Các phương thức truyền dữ liệu trong FTP

Khi kênh dữ liệu đã được thiết lập xong giữa Server-DTP với User-DTP, dữ liệu sẽ được truyền trực tiếp từ phía client tới phía server, hoặc ngược lại, dựa theo các lệnh được sử dụng. Do thông tin điều khiển được gửi đi trên kênh điều khiển, nên toàn bộ kênh dữ liệu có thể được sử dụng để truyền dữ liệu. (Tất nhiên, hai kênh logic này được kết hợp với nhau ở lớp dưới cùng với tất cả các kết nối TCP/UDP khác giữa hai thiết bị, do đó điều này không hẳn đã cải thiện tốc độ truyền dữ liệu so với khi truyền trên chỉ một kênh – nó chỉ làm cho hai việc truyền dữ liệu và điều khiển trở nên độc lập với nhau mà thôi)

FTP có ba phương thức truyền dữ liệu, nêu lên cách mà dữ liệu được truyền từ một thiết bị tới thiết bị khác trên một kênh dữ liệu đã được khởi tạo, đó là: *stream mode*, *block mode*, và *compressed mode*.

##### *Stream mode*

Trong phương thức này, dữ liệu được truyền đi dưới dạng các byte không cấu trúc liên tiếp. Thiết bị gửi chỉ đơn thuần đẩy luồng dữ liệu qua kết nối TCP tới phía nhận. Không có một trường tiêu đề nhất định được sử dụng trong phương thức này



làm cho nó khá khác so với nhiều giao thức gửi dữ liệu rời rạc khác. Phương thức này chủ yếu dựa vào tính tin cậy trong truyền dữ liệu của TCP. Do nó không có cấu trúc dạng header, nên việc báo hiệu kết thúc file sẽ đơn giản được thực hiện việc phía thiết bị gửi ngắt kênh kết nối dữ liệu khi đã truyền xong.

Trong số ba phương thức, stream mode là phương thức được sử dụng nhiều nhất trong triển khai FTP thực tế. Có một số lý do giải thích điều đó. Trước hết, nó là phương thức mặc định và đơn giản nhất, do đó việc triển khai nó là dễ dàng nhất. Thứ hai, nó là phương pháp phổ biến nhất, vì nó xử lý với các file đều đơn thuần như là xử lý dòng byte, mà không để ý tới nội dung của các file. Thứ ba, nó là phương thức hiệu quả nhất vì nó không tốn một lượng byte “overload” để thông báo header.

### *Block mode*

Đây là phương thức truyền dữ liệu mang tính quy chuẩn hơn, với việc dữ liệu được chia thành nhiều khối nhỏ và được đóng gói thành các FTP blocks. Mỗi block này có một trường header 3 byte báo hiệu độ dài, và chứa thông tin về các khối dữ liệu đang được gửi. Một thuật toán đặc biệt được sử dụng để kiểm tra các dữ liệu đã được truyền đi và để phát hiện, khởi tạo lại đối với một phiên truyền dữ liệu đã bị ngắt.

### *Compressed mode*

Đây là một phương thức truyền sử dụng một kỹ thuật nén khá đơn giản, là “run-length encoding” – có tác dụng phát hiện và xử lý các đoạn lặp trong dữ liệu được truyền đi để giảm chiều dài của toàn bộ thông điệp. Thông tin khi đã được nén, sẽ được xử lý như trong block mode, với trường header. Trong thực tế, việc nén dữ liệu thường được sử dụng ở những chỗ khác, làm cho phương thức truyền kiểu compressed mode trở nên không cần thiết nữa. Ví dụ: nếu bạn đang truyền đi một file qua internet với modem tương tự, modem của bạn thông thường sẽ thực hiện việc nén ở lớp 1; các file lớn trên FTP server cũng thường được nén sẵn với một số định dạng như ZIP, làm cho việc nén tiếp tục khi truyền dữ liệu trở nên không cần thiết.

## PHẦN IV: PHÂN LOẠI

FTP là một phương pháp truyền tập tin có truyền thống phi bảo an (không an toàn), vì theo như bản thiết kế gốc đặc tả của FTP, không có cách nào có thể truyền tải dữ liệu dưới hình thức mật mã hóa được.

Ảnh hưởng này có nghĩa là, phần lớn các cài đặt của mạng lưới truyền thông, tên người dùng, mật khẩu, dòng lệnh FTP và tập tin được truyền tải, đều có thể bị người khác trên cùng một mạng lưới, "ngửi" hoặc quan sát, dùng phần mềm phân tích giao thức (protocol analyzer) (hoặc còn gọi là "dụng cụ ngửi dữ liệu", tiếng Anh là "sniffer"). Nên chú ý rằng đây là vấn đề thường thấy ở các giao thức của Internet được thiết kế trước khi SSL (Secure Sockets Layer) ra đời (tạm dịch là giao thức "tàng kết nối bảo mật"), như HTTP, SMTP và Telnet. Giải pháp thường thấy, đối với vấn đề này, là dùng SFTP (Secure Shell File Transfer Protocol - tạm dịch là "giao thức truyền tập tin dùng trình bao bảo mật"), một giao thức dựa trên nền của SSH, hoặc trên FTPS (FTP over SSL). SFTP là FTP được cộng thêm chức năng mã hoá dữ liệu của SSL hoặc TLS (Transport Layer Security - tạm dịch là "Bảo mật tầng giao vận").

*Các giao thức tương tự như FTP:*

- FTPFS
- FTPS, FTP chạy trên nền SSL
- SFTP (Simple File Transfer Protocol), một giao thức có tính lịch sử RFC 913
- Giao thức truyền tập tin dùng trình bao bảo mật (SSH file transfer protocol - SFTP), một giao thức chạy trên nền của SSH (Secure SHell - trình bao bảo mật)
- TFTP (Trivial File Transfer Protocol - Giao thức truyền tập tin tầm thường)

*Tìm hiểu thêm:*

### 1. SFTP

SFTP (viết tắt của từ Secure File Transfer Protocol, hoặc SSH File Transfer Protocol) là một giao thức mạng giúp bạn có thể upload hoặc download dữ liệu trên

máy chủ. Bạn cũng có thể sử dụng giao thức này để sửa, tạo hoặc xóa các tập tin và thư mục trên máy chủ Linux.

Bởi vì SFTP sử dụng giao thức SSH để kết nối nên các dữ liệu bạn di chuyển thông qua giao thức này cũng sẽ được mã hóa để bảo mật dữ liệu tốt hơn.

Giao thức Secure FTP khắc phục vấn đề lộ bí mật trên bằng cách mã hóa tất cả thông tin được gửi cho :

Cho CLIENT SFTP.( Dreamweaver ).

Và gửi cho Server FTP.

## **2. FTPS**

FTPS là FTP trên nền SSL để bảo mật. Nó sử dụng một kênh điều khiển và mở các kết nối mới cho việc truyền dữ liệu. Việc sử dụng SSL đòi hỏi có chứng thư số.

- Ưu điểm: Nổi tiếng và được sử dụng rộng rãi. Các thông tin liên lạc có thể được đọc và hiểu bởi một người. Cung cấp dịch vụ để chuyển tập tin từ máy chủ đến máy chủ. SSL/TLS có cơ chế xác thực tốt (tính năng chứng nhận thư X.509). FTP và SSL/TLS được hỗ trợ xây dựng vào các nền tảng truyền thông trên internet.
- Nhược điểm: Không có một định dạng thống nhất danh sách thư mục. Cần phải có một kênh dữ liệu thứ cấp, do vậy, gây khó khăn khi sử dụng phía sau tường lửa. Không xác định một tiêu chuẩn cho bộ tên tập tin ký tự (mã hóa); Không phải hỗ trợ tất cả các máy chủ FTP SSL/TLS. Không có tiêu chuẩn để có được và thay đổi tập tin hoặc các thuộc tính thư mục.

## **3. SFTP**

SFTP (SSH File Transfer Protocol/Secure File Transfer Protocol) được thiết kế như một phần mở rộng của SSH để cung cấp khả năng truyền tập tin, do đó, nó thường chỉ sử dụng cổng SSH cho cả dữ liệu và kiểm soát.

- Ưu điểm: Có một nền tảng tiêu chuẩn tốt, trong đó, xác định chính xác (nếu không phải tất cả) các khía cạnh của hoạt động. Chỉ có một kết nối (không cần kết nối dữ liệu). Các kết nối được luôn được an toàn. Danh sách thư mục đồng

bộ và máy có thể đọc được. Các giao thức bao gồm các hoạt động cho phép và thuộc tính thao tác, khóa tập tin và nhiều chức năng khác.

- **Nhược điểm:** Các thông tin liên lạc là nhị phân và không thể đăng nhập. SSH là khó quản lý và chứng nhận. Các tiêu chuẩn xác định là tùy chọn hoặc khuyến nghị, dẫn đến vấn đề tương thích nhất định giữa các phần mềm khác nhau từ các nhà cung cấp khác nhau. Không sao chép từ máy chủ đến máy chủ và các hoạt động loại bỏ thư mục đệ quy. Không được hỗ trợ SSH/SFTP trong các nền tảng trên Internet.

## **4. TFTP**

Là 1 giao thức rất đơn giản để truyền nhận files. Sự đơn giản thể hiện ở ngay cái tên của nó: Trivial File Transfer Protocol (TFTP). Đầu tiên được thiết kế bởi Noel Chiappa, sau đó được nhiều người chỉnh sửa, bổ sung. Nó dùng cơ chế ACK và truyền lại (resending) theo ý tưởng của TCP, cơ chế kiểm lỗi thì dựa trên cơ chế thông điệp trong EFTP của PARC. TFTP đã được hiện thực trên nền tảng của UPD vì vậy nó có thể dùng để di chuyển file trên các máy trên các mạng khác nhau có hiện thực UDP. Nhưng điều này không loại trừ khả năng hiện thực TFTP trên các protocol datagram khác. Nó được thiết kế để hiện thực nhỏ gọn và đơn giản, do đó FTP là giao thức bổ sung các thiếu sót của TFTP.

Một ví dụ về ứng dụng của TFTP: Boot từ xa các host diskless, TFTP dùng để download các file cấu hình cơ bản từ thư mục /boot (lúc này không cần cơ chế nhận dạng).

Nếu dùng không cẩn thận, TFTP sẽ cho phép bất cứ ai download bất cứ file gì trên hệ thống của bạn.

TFTP (chỉ) có thể:

- Đọc và ghi file (hoặc mail) đến/đi từ 1 server từ xa.

TFTP không thể:

- Liệt kê các thư mục
- Chưa có cơ chế nhận dạng user (nên không có bảo vệ bằng password như FTP)

Hiện tại có 3 kiểu transfer được hỗ trợ:

- *Netascii*: netascii là mã ASCII được chỉnh sửa theo đặc tả “Telnet Protocol Specification” - đây là ASCII 8 bit
- *Octet*: 8 bit thô
- *Mail*: là các kí tự netascii được gửi đến 1 user chứ không phải 1 file ( kiểu truyền mail này đã cổ xưa và không nên được hiện thực hay dùng)

Ngoài ra 2 host có thể thoả thuận với nhau để xác định các kiểu transfer khác.

## PHẦN V: MÔ PHỎNG GIAO THỨC

### 1. Phần mềm mô phỏng:

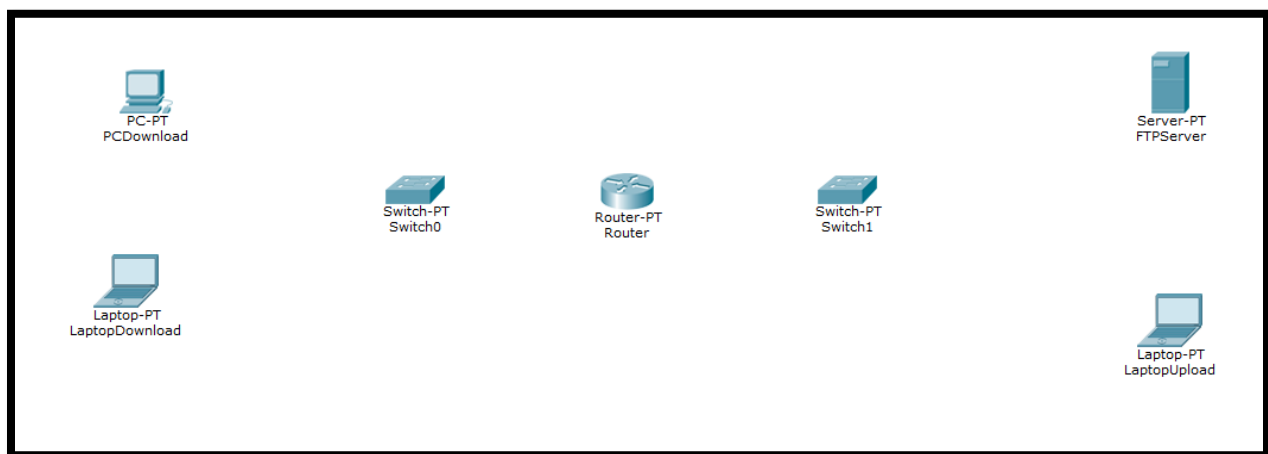
Phần mềm Cisco Packet Tracer là một công cụ nổi tiếng của hãng Cisco dùng để mô phỏng các giao thức giao thức trên mạng. Phiên bản mới nhất của Packet Tracer là 7.1. Tuy nhiên các tính năng mới của 7.1 không khác so với 7.0 nhiều nên trong bài tìm hiểu lần này, nhóm em xin được dùng phiên bản 7.0 để mô phỏng cách thức hoạt động của giao thức FTP.

### 2. Cài đặt và mô phỏng:

#### 2.1 Khởi tạo các đối tượng:

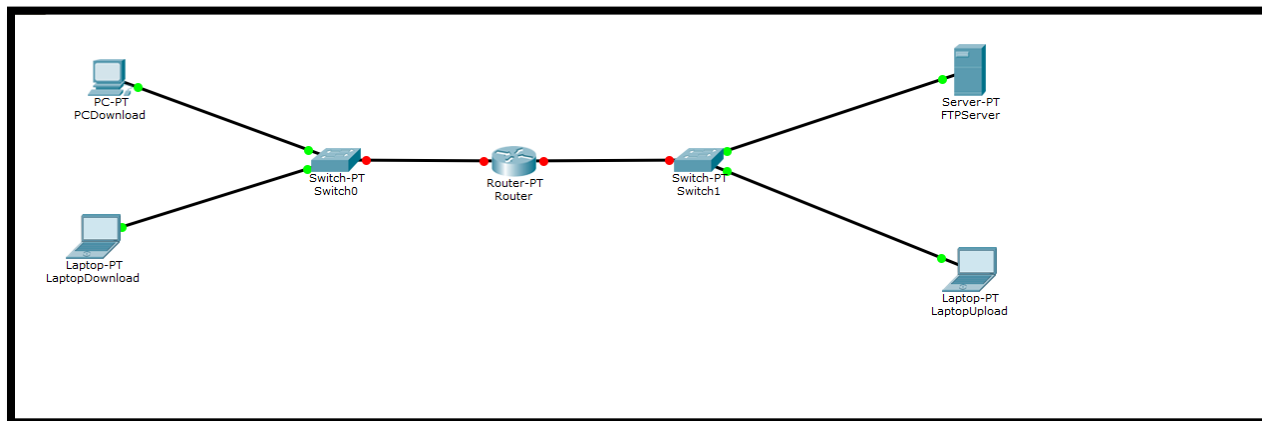
Các đối tượng trong bài mô phỏng bao gồm :

- 1 PC tên PCDownload
- 2 Laptop tên LaptopDownload và LaptopUpload
- 2 Switch tên Switch0 và Switch 1
- 1 Router tên Router
- 1 Server tên FTPServer



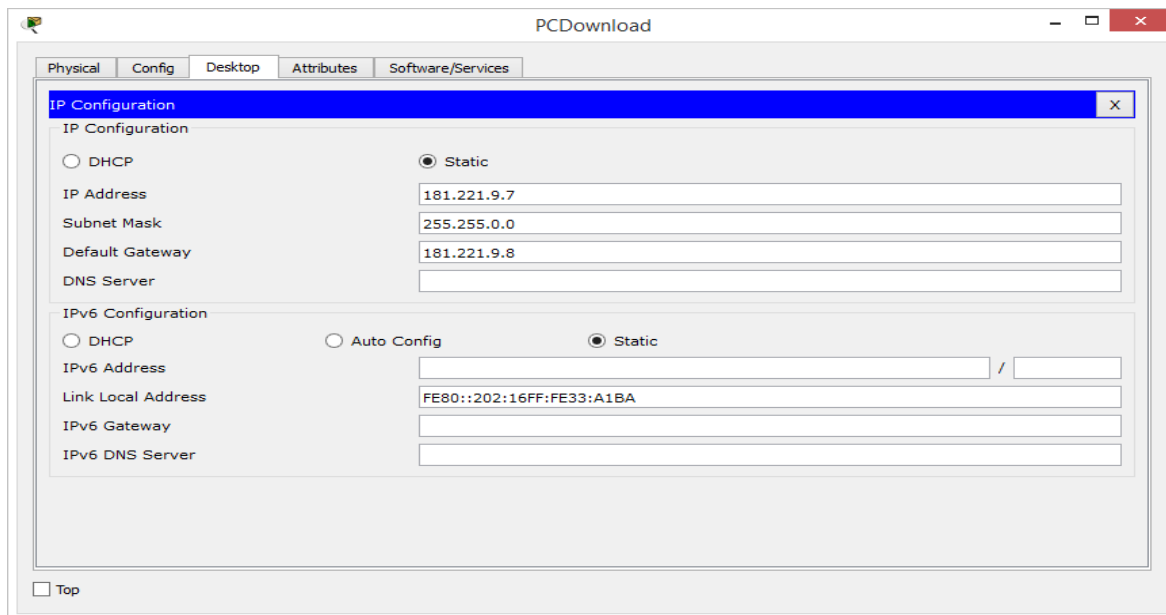
## 2.2. Thiết lập các kết nối:

Các thiết bị được nối với nhau thông qua đường cáp đồng thẳng đối xứng (Copper Straight-Through), truyền tải bằng Fast Ethernet Mô tả như hình dưới:

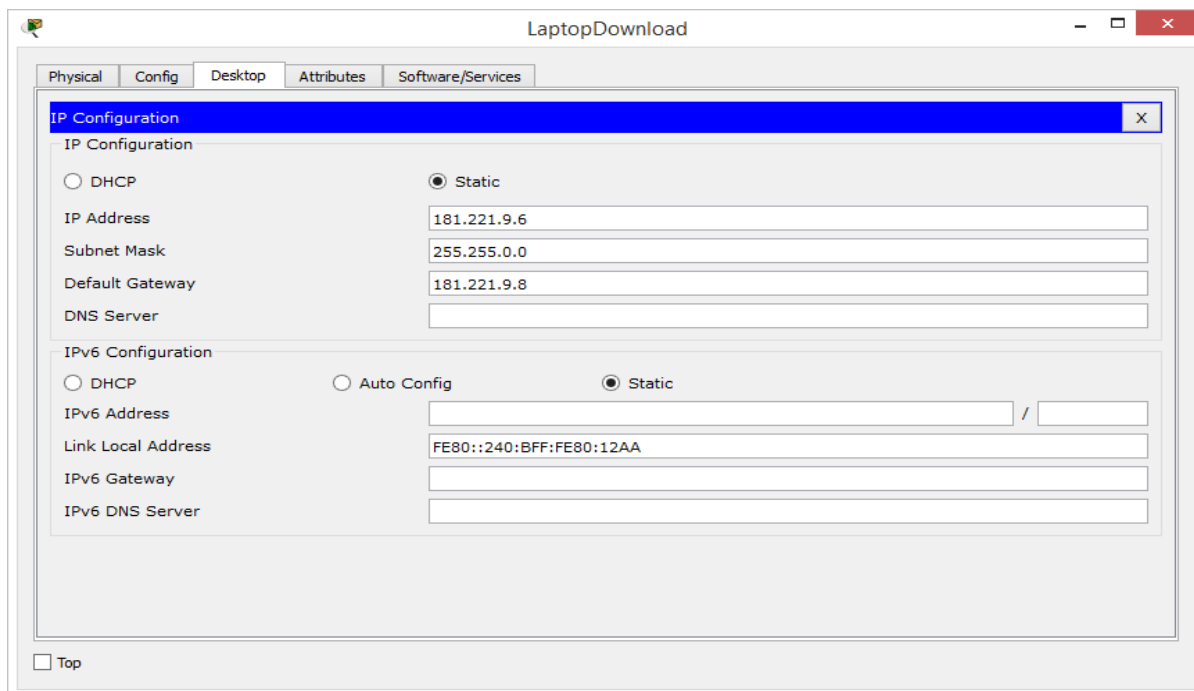


## 2.3. Thiết lập thuộc tính các đối tượng:

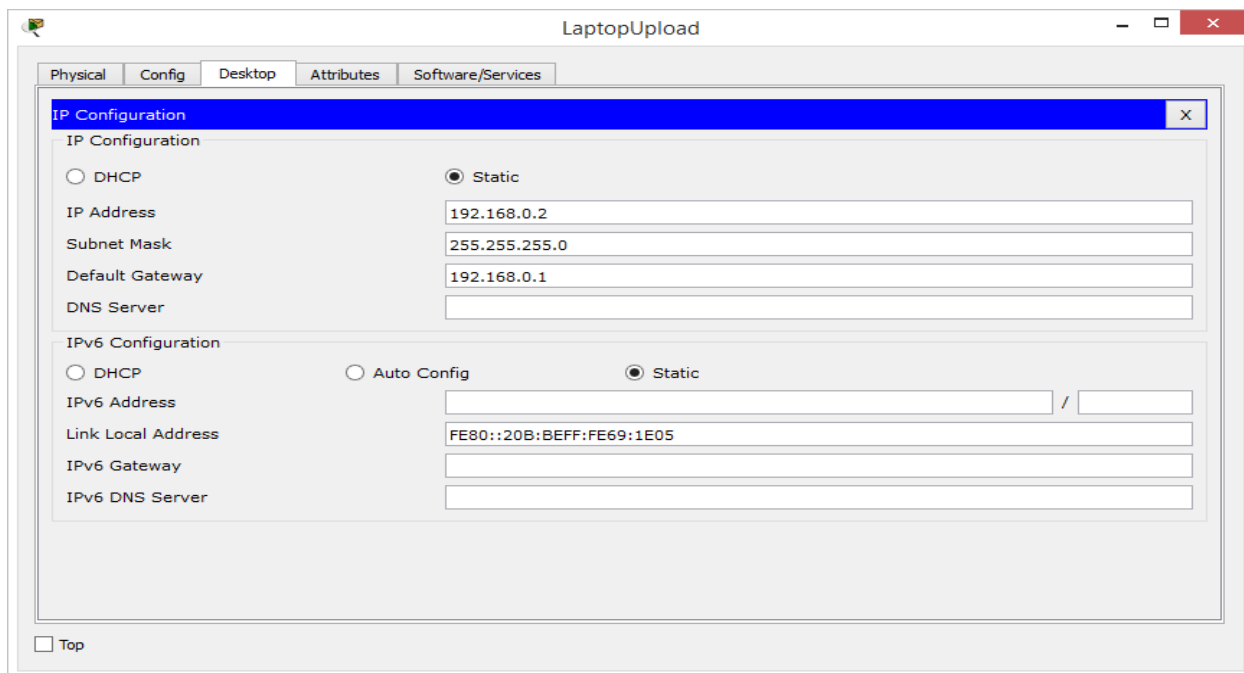
- Máy PC Download và LaptopDownload cùng một mạng nên cùng HostID, 180.1.22 và subnet mask 255.255.0.0 gateWay: 192.168.1.1.
  - PCDownload nhận địa chỉ IP 181.221.9.7



- LaptopDownload nhận địa chỉ IP 181.221.9.6

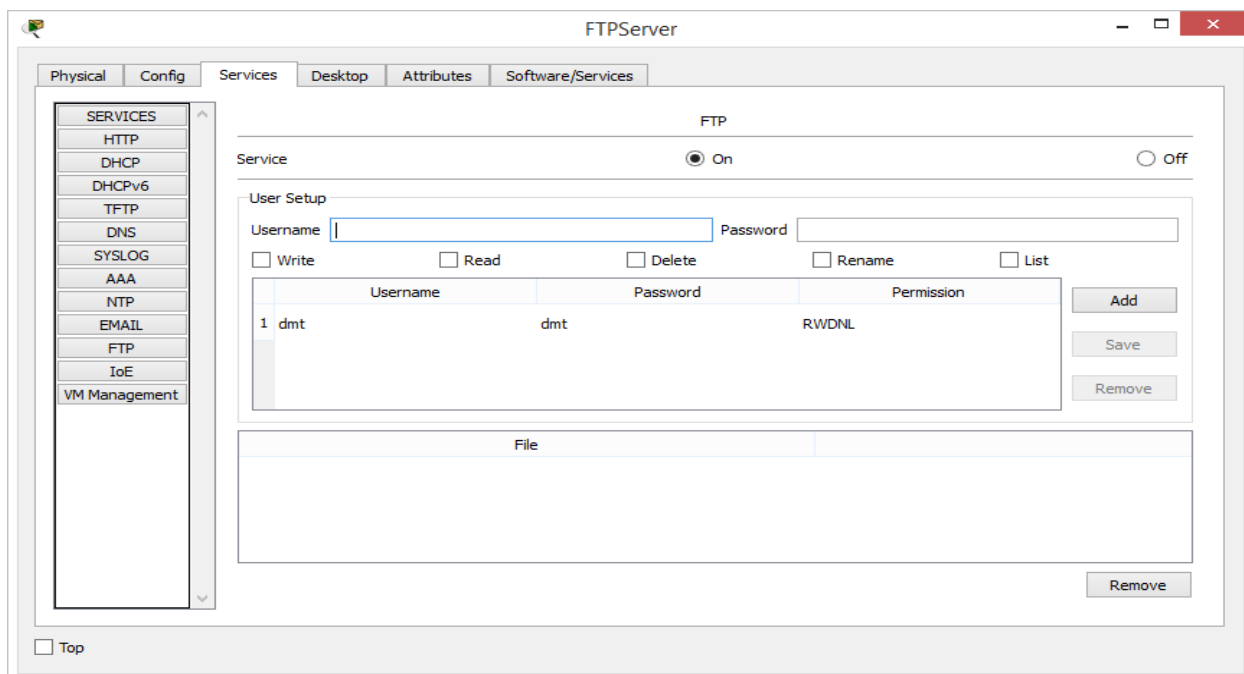


- Máy chủ FTPServer và LaptopUpload cùng 1 mạng nên cùng HostID:  
192.168.0 subnetmask 255.255.255.0 gateway 192.168.0.1
  - Máy LaptopUpload nhận địa chỉ 192.168.0.2

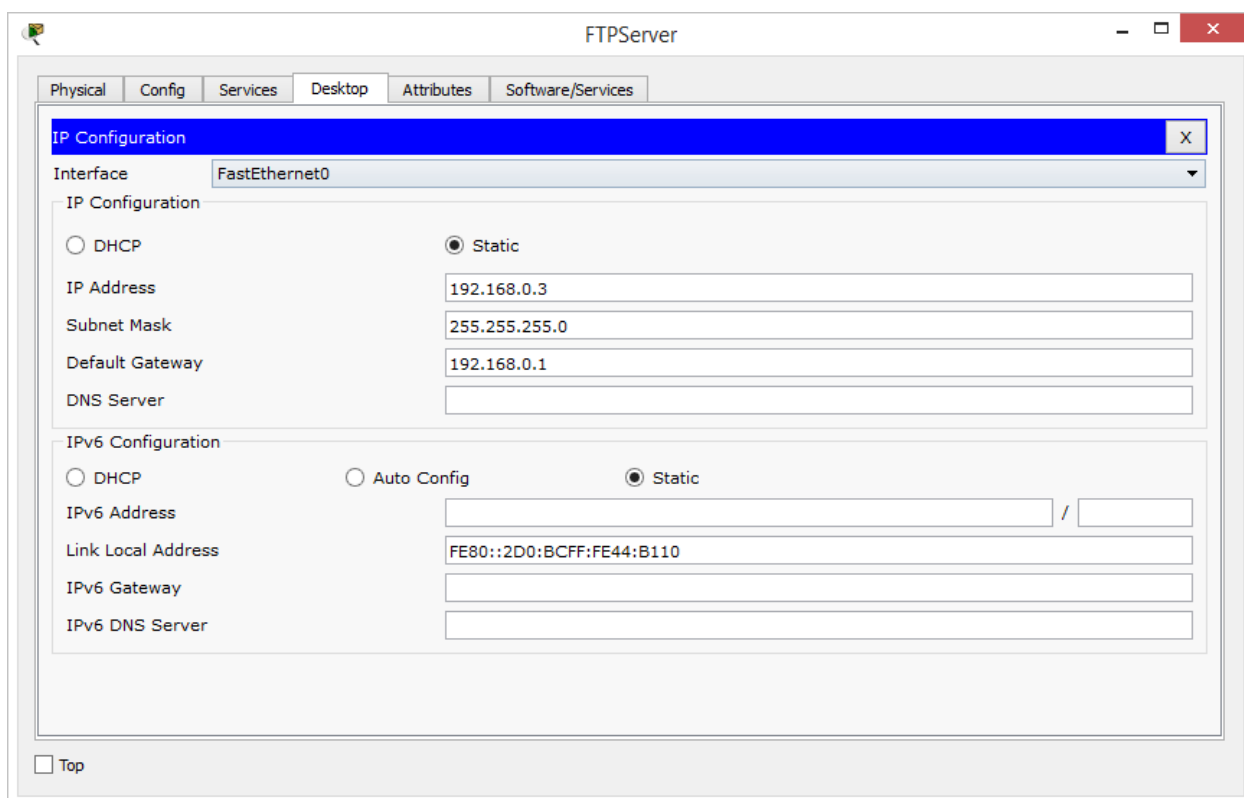


- Thiết lập cấu hình máy chủ FTPServer

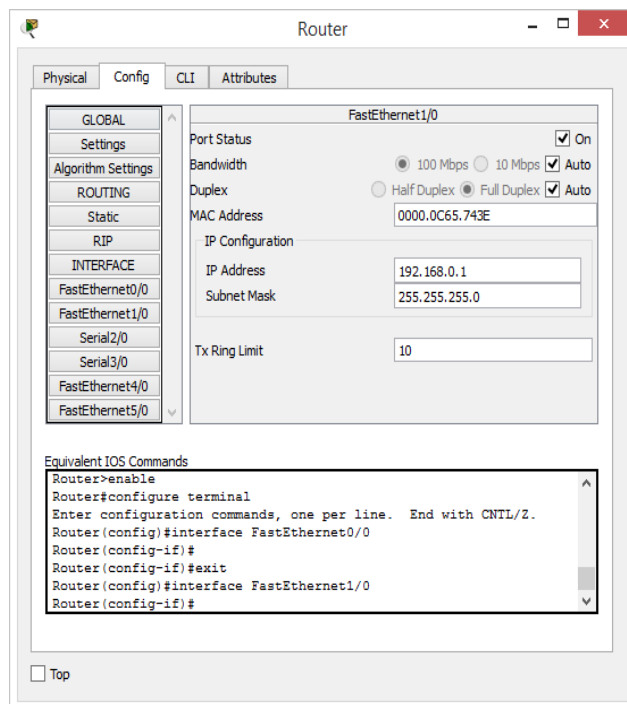
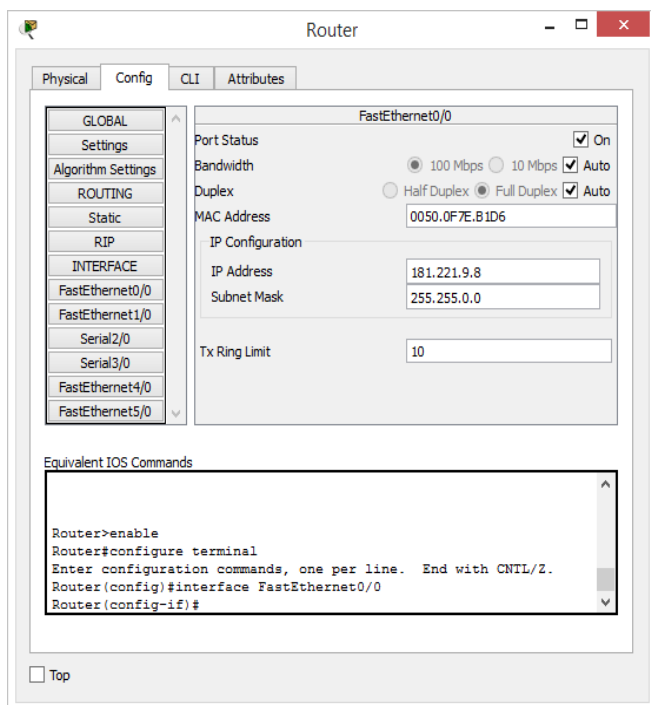




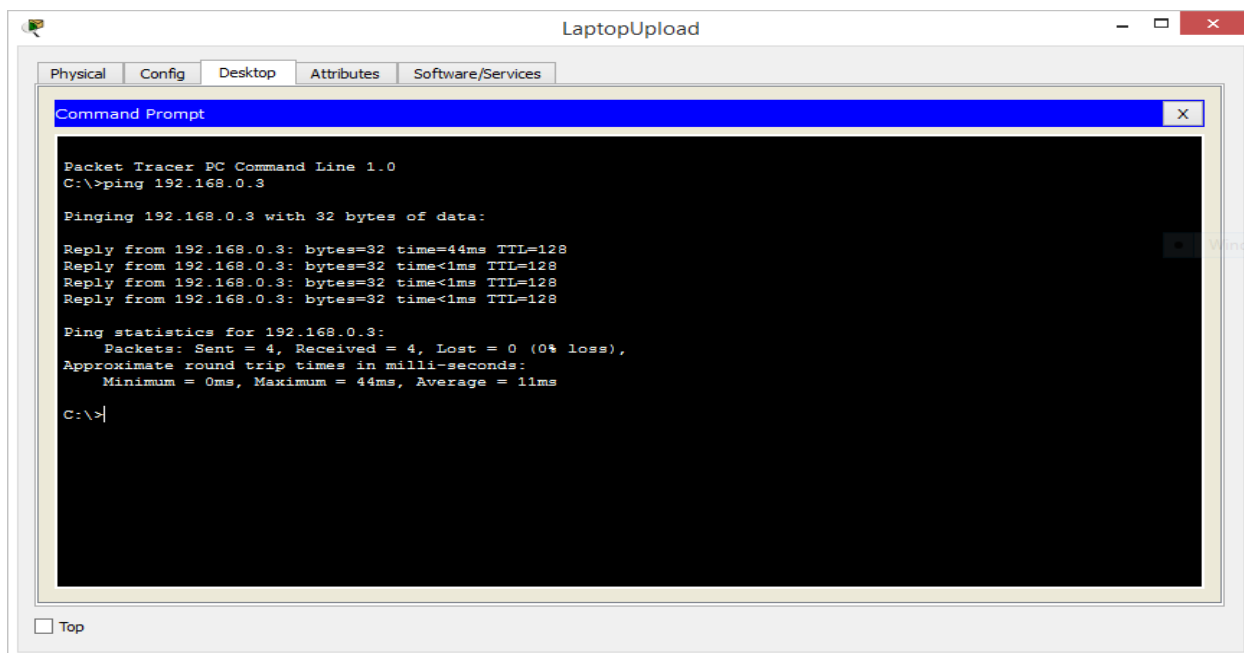
Gồm 1 tài khoản dmt/dmt với tất cả các quyền đọc, viết, xóa, sửa, liệt kê và không tồn tại tập tin nào trước. Địa chỉ IP như hình vẽ:

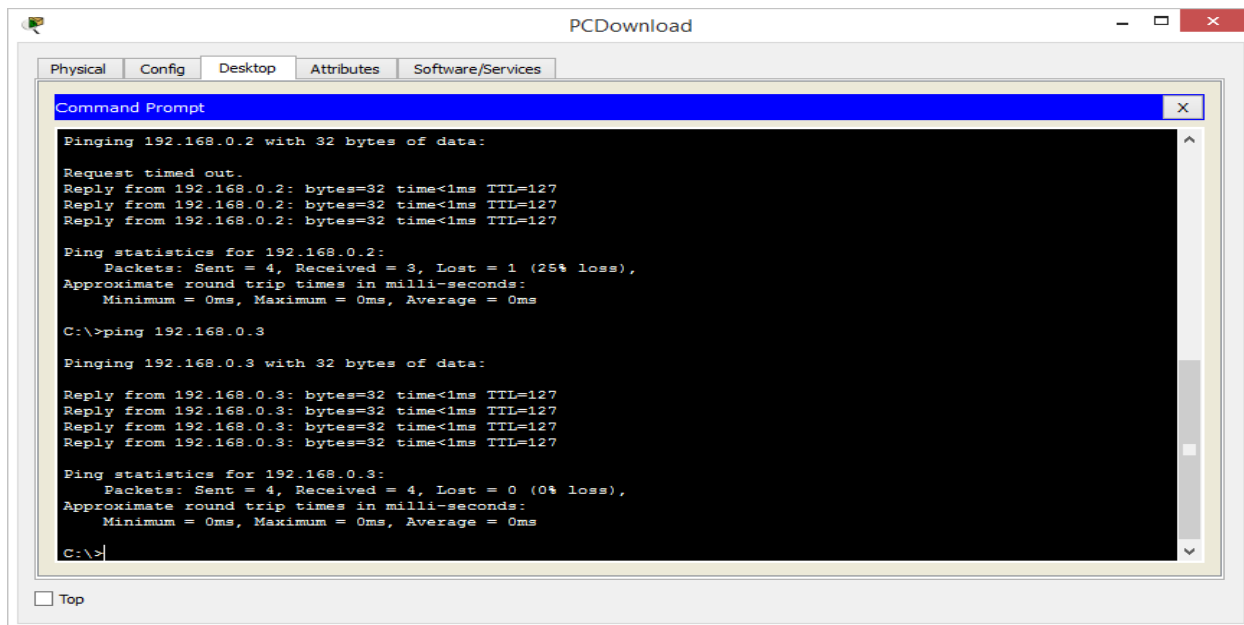
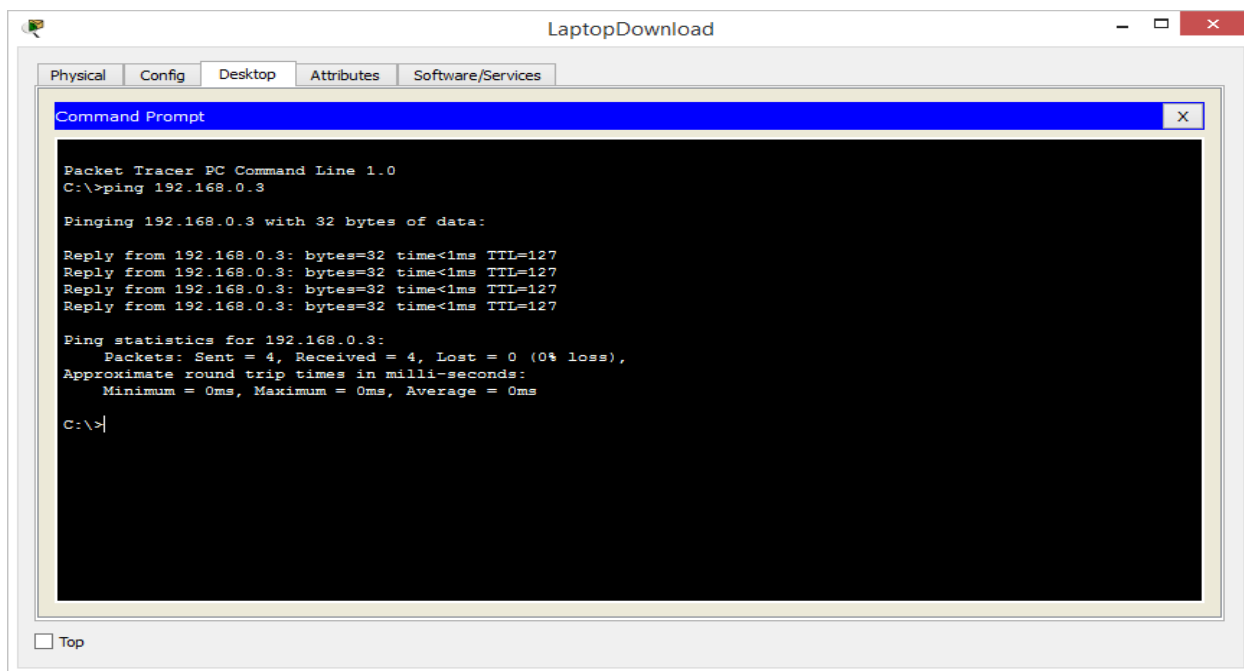


- Router kết nối 2 mạng



## 2.4. Test kết nối tới FTPServer trên cmd với lệnh ping

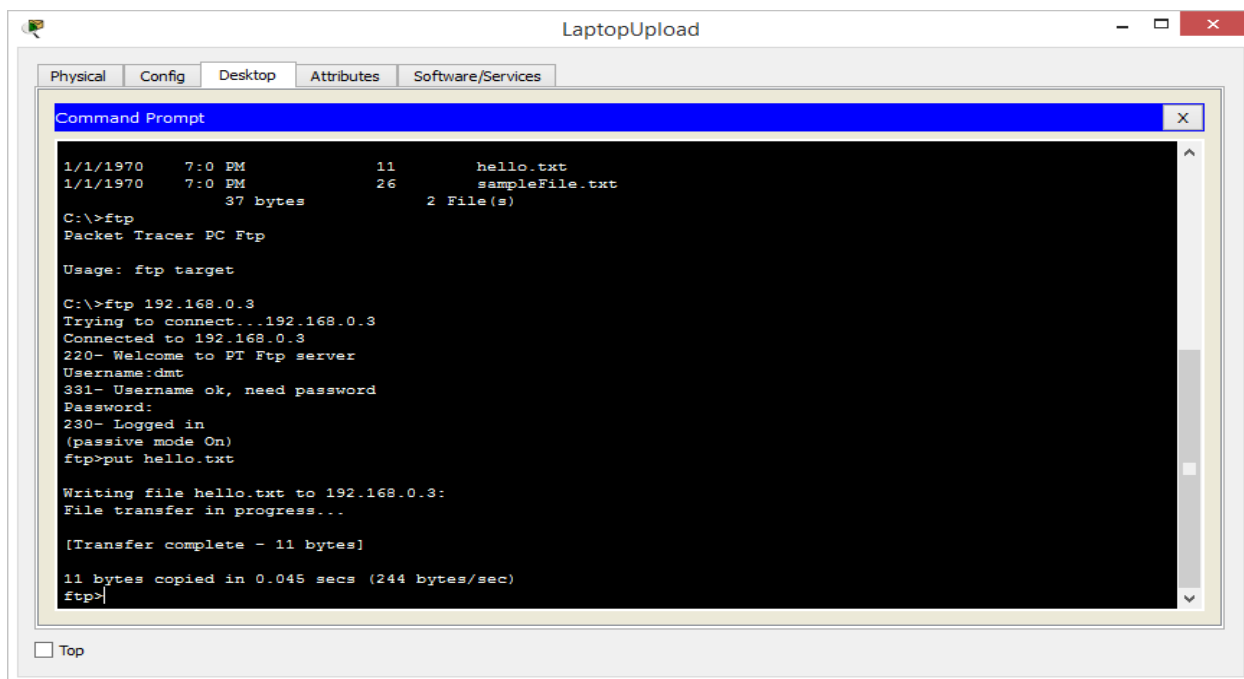




⇒ Các gói tin được gửi với độ mất mát 0% đến Server chứng tỏ thiết lập kết nối đã thành công.

### 2.5. Tải tệp tin lên / từ FTPServer:

Tại máy LaptopUpload, dùng lệnh [ftp 192.168.0.3](#), đăng nhập bằng tài khoản dmt/dmt, tải thử tệp hello.txt vừa tạo trên máy này lên FTPServer bằng lệnh put hello.txt



```
Command Prompt
1/1/1970  7:0 PM          11      hello.txt
1/1/1970  7:0 PM          26      sampleFile.txt
                               37 bytes      2 File(s)

C:\>ftp
Packet Tracer PC Ftp

Usage: ftp target

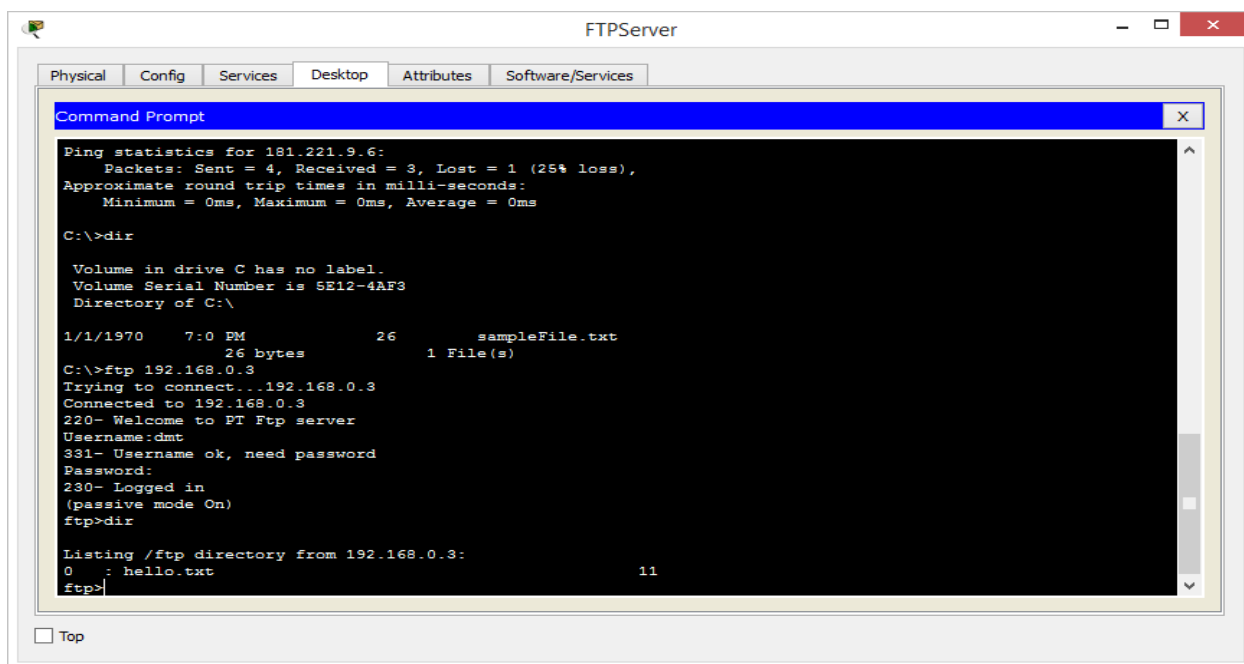
C:\>ftp 192.168.0.3
Trying to connect...192.168.0.3
Connected to 192.168.0.3
220- Welcome to PT Ftp server
Username:dmt
331- Username ok, need password
Password:
230- Logged in
(passive mode On)
ftp>put hello.txt

Writing file hello.txt to 192.168.0.3:
File transfer in progress...

[Transfer complete - 11 bytes]

11 bytes copied in 0.045 secs (244 bytes/sec)
ftp>
```

Kiểm tra tệp hello.txt đã có trên máy chủ bằng lệnh dir



```
FTPServer
Physical  Config  Services  Desktop  Attributes  Software/Services

Command Prompt

Ping statistics for 192.168.0.3:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>dir

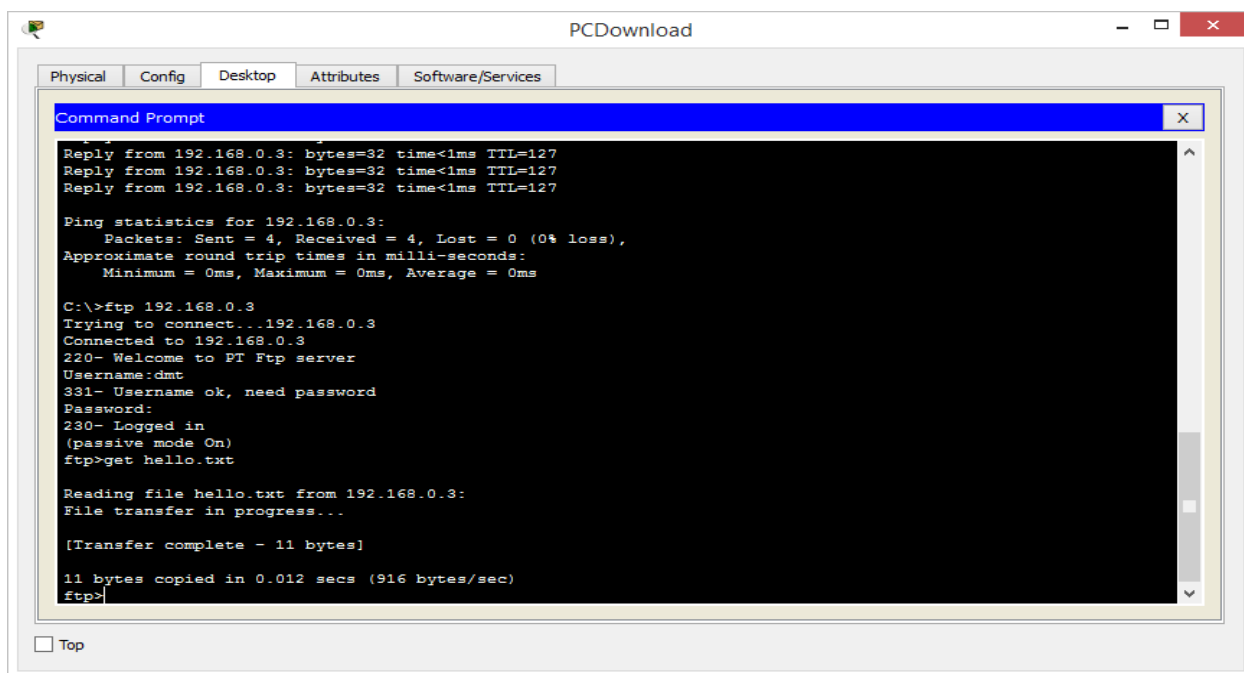
Volume in drive C has no label.
Volume Serial Number is 5E12-4AF3
Directory of C:\

1/1/1970  7:0 PM          26      sampleFile.txt
                               26 bytes      1 File(s)

C:\>ftp 192.168.0.3
Trying to connect...192.168.0.3
Connected to 192.168.0.3
220- Welcome to PT Ftp server
Username:dmt
331- Username ok, need password
Password:
230- Logged in
(passive mode On)
ftp>dir

Listing /ftp directory from 192.168.0.3:
0       : hello.txt                               11
ftp>
```

Tại máy PCDownload và LaptopDownload ta có thể tải tệp này về bằng lệnh get hello.txt



```
PCDownload
Physical Config Desktop Attributes Software/Services
Command Prompt
Reply from 192.168.0.3: bytes=32 time<1ms TTL=127
Reply from 192.168.0.3: bytes=32 time<1ms TTL=127
Reply from 192.168.0.3: bytes=32 time<1ms TTL=127

Ping statistics for 192.168.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>ftp 192.168.0.3
Trying to connect...192.168.0.3
Connected to 192.168.0.3
220- Welcome to PT Ftp server
Username:dmt
331- Username ok, need password
Password:
230- Logged in
(passive mode On)
ftp>get hello.txt

Reading file hello.txt from 192.168.0.3:
File transfer in progress...

[Transfer complete - 11 bytes]
11 bytes copied in 0.012 secs (916 bytes/sec)
ftp>
```

⇒ Như vậy, chúng ta chứng tỏ việc thiết lập giao thức FTP trên máy chủ FTPServer đã thành công. Các tệp tin từ các máy tính cá nhân có thể tải và lấy dữ liệu trên máy chủ về một khi có tên đăng nhập và mật khẩu.

## PHẦN VI: PHÂN TÍCH VÀ KẾT LUẬN

### 1. Phân tích ưu nhược điểm của FTP:

#### 1. Ưu điểm

Sử dụng FTP người ta có thể download hay upload mỗi lần được nhiều file (Internet Explorer chỉ cho nhiều nhất 2 file).

Trên server FTP có tính năng bảo mật rất cao. Những dữ liệu cá nhân riêng mà mỗi user đưa vào folder tạo riêng được hoàn toàn bảo mật, không có một user khác nào trong FTP có quyền can thiệp hay nhìn thấy folder của người khác.

Cách sử dụng folder riêng trên FTP cũng như cách sử dụng folder trong Windows. Khi chúng ta muốn đưa dữ liệu riêng của mình lên FTP thì chỉ cần thực hiện thao tác Copy dữ liệu mà ta muốn đưa lên FTP rồi đưa vào folder tạo riêng rồi thực hiện thao tác Paste.

#### 2. Nhược điểm

FTP có các tùy chọn bảo vệ bằng mật khẩu, nhưng chính các tên truy nhập và mật khẩu lại được gửi qua Internet mà không được mã hoá, nên kẻ xấu hoàn toàn có thể đánh cắp chìa khoá và sau đó là thông tin. Vấn đề lớn hơn nữa là khả năng của FTP cho phép mọi người truy nhập tự do. Mục đích ban đầu là để khuyến khích chia sẻ, nhưng mặt khác điều này cũng dễ gây ra việc vô tình mất những thông tin, dữ liệu riêng tư hoặc nhạy cảm.

Mật khẩu và nội dung của tập tin được truyền qua đường dây ở thể dạng văn bản thường (clear text), và vì vậy chúng có thể bị chặn và nội dung bị tiết lộ cho những kẻ nghe trộm. Hiện nay, người ta có những cải tiến để né tránh được nhược điểm này.

Cần phải có nhiều kết nối TCP/IP: một dòng dành riêng cho việc điều khiển kết nối, một dòng riêng cho việc truyền tập tin lên, truyền tập tin xuống, hoặc liệt kê thư mục. Các phần mềm bức tường lửa cần phải được cài đặt thêm những logic mới, để có thể lường trước được những kết nối của FTP.

Việc thanh lọc giao thông FTP bên trình khách, khi nó hoạt động ở chế độ năng động, dùng bức tường lửa, là một việc khó làm, vì trình khách phải tùy ứng mở một cổng mới để tiếp nhận đòi hỏi kết nối khi nó xảy ra. Vấn đề này phần lớn được giải quyết bằng cách chuyển FTP sang dùng ở chế độ bị động.

Người ta có thể lạm dụng tính năng ủy quyền, được cài đặt sẵn trong giao thức, để sai khiến máy chủ gửi dữ liệu sang một cổng tùy chọn ở một máy tính thứ ba.

FTP là một giao thức có tính trì trệ rất cao (high latency). Sự trì trệ gây ra do việc nó bắt buộc phải giải quyết một số lượng lớn các dòng lệnh khởi đầu một phiên truyền tải.

Phần nhận không có phương pháp để kiểm chứng tính toàn vẹn của dữ liệu được truyền sang. Nếu kết nối truyền tải bị ngắt giữa chừng thì không có cách gì, trong giao thức, giúp cho phần nhận biết được rằng, tập tin nhận được là hoàn chỉnh hay còn vẫn còn thiếu sót. Sự hỗ trợ bên ngoài, như việc dùng kiểm tra tổng MD5, hoặc dùng kiểm độ dư tuần hoàn (cyclic redundancy checking) là một việc cần thiết.

## 2. Kết luận:

“FTP là một protocol trong mô hình TCP/IP được dùng để truyền các file giữa các máy. FTP cho phép truyền nhận file và quản lý trực tuyến. FTP không cho phép truy xuất một máy khác để thực thi chương trình, nhưng nó rất tiện lợi cho việc thao tác với file”

*Qua bài tập lớn lần này, chúng em đã có cơ hội được tìm hiểu giao thức File Transfer Protocol, là một giao thức rất phổ biến và quan trọng cho đến hiện nay. Nhóm đã tìm hiểu cách thức hoạt động của giao thức FTP, giới thiệu các phiên bản khác nhau trong mô hình Client – Server. Phân loại và mô phỏng được phân tích khá rõ ràng phần trên. Ở đây, nên nhóm em chỉ tập trung làm những chức năng cơ bản, những phần nâng cao hơn thì cần thêm thời gian ở các môn học chuyên sâu.*

---

# DANH MỤC THAM KHẢO

---

1. Abhay Bhushan, RFC 354 - File Transfer Protocol, MIT- MAC, July 8 – 1972
  2. Nancy J. Neigus, RFC 542 - File Transfer Protocol, Cambridge - Mass, Aug. 12 – 1973
  3. A. McKenzie, RFC 454 - File Transfer Protocol, BBN, Feb. 16 - 1973
  4. J. Postel & J. Reynolds, RFC 959 - THE TFTP PROTOCOL (REVISION 2), Oct. – 1985
  5. K. Sollins, RFC 1350 - File Transfer Protocol, July – 1992
  6. Nguyễn Thúc Hải, Mạng máy tính và các hệ thống thông tin, 1999
- Và một số tài liệu internet khác ...

***Chúng em cũng xin phép được chân thành cảm ơn cô  
Ngô Quỳnh Thu đã hướng dẫn chúng em làm bài tập  
lớn cùng như hoàn thành môn học mạng máy tính  
này.***