**Giao thức truyền file FTP**

Giao thức FTP (File Transfer Protocol) là một giao thức trao đổi file khá phổ biến hiện nay, nó được sử dụng nhiều nhất vào mục đích truyền tải dữ liệu. bài viết này mình sẽ giới thiệu cho các bạn đặc điểm, nguyên lý hoạt động đến những hoạt động xảy ra trong giao thức.

**1. Giới thiệu về FTP**

FTP được tạo ra với mục tiêu chính là cho phép những máy tính trên 1 mạng gián tiếp có thể dễ dàng chuyển file từ máy tính này sang máy tính khác. Giống như hầu hết các giao thức TCP/IP, nó dựa trên mô hình client-server, với một FTP client trên máy tính người dùng tạo ra một kết nối đến một FTP server để gửi và lấy các tập tin đến và đi từ các máy chủ. Những mục tiêu chính của FTP là để làm cho chuyển tập tin đơn giản, và che chắn người dùng từ việc thực hiện chi tiết về làm thế nào các tập tin đang thực sự di chuyển từ một nơi khác. Để kết thúc, FTP được thiết kế để tự động đối phó với nhiều vấn đề có khả năng phát sinh do có sự khác biệt về định dạng tập tin được lưu trữ trên hệ thống khác nhau.

Để đảm bảo rằng các tập tin được gửi và nhận được mà không làm mất dữ liệu có thể hỏng chúng, FTP sử dụng giao thức điều khiển truyền dẫn tin cậy tại tầng giao vận. Một hệ thống xác thực được sử dụng để đảm bảo rằng máy khách ủy quyền chỉ được cho phép để truy cập vào một máy chủ. Cùng lúc đó, một tính năng đôi khi được gọi là ẩn danh FTP cho phép một tổ chức mong muốn nó để thiết lập một máy chủ tổng hợp thông tin để cung cấp các tập tin cho bất cứ ai có thể lấy chúng.

Sau khi một kết nối TCP được thiết lập, một điều khiển kết nối FTP được tạo ra. Nội bộ lệnh FTP được truyền qua kết nối logic này dựa trên định dạng quy tắc được thiết lập bởi các giao thức Telnet. Mỗi lệnh được gửi bởi các khách hàng sẽ nhận được một trả lời từ máy chủ để thông báo nó đã thành công hay thất bại. Kết nối dữ liệu được thiết lập cho mỗi dữ liệu cá nhân được chuyển. FTP hỗ trợ cả kết nối dữ liệu bình thường hoặc thụ động, cho phép các máy chủ hoặc khách hàng đều có thể bắt đầu kết nối dữ liệu. Nhiều loại dữ liệu và các loại tệp được hỗ trợ để cho phép sự linh hoạt với nhiều loại hình dịch vụ đưa đón.

Giao diện giữa một người sử dụng FTP và các giao thức được cung cấp dưới dạng một tập các lệnh tương tác người dùng. Sau khi thiết lập kết nối, và hoàn tất xác thực hai lệnh cơ bản có thể được sử dụng để gửi hoặc nhận tệp.Ngoài ra những lệnh bổ sung được cung cấp để quản lý kết nối FTP, cũng như để thực hiện hỗ trợ chức năng chẳng hạn như danh sách nội dung của một thư mục hoặc xóa hoặc đổi tên tập tin. Những năm gần đây, đồ họa thực của FTP đã được tạo ra để cho phép người sử dụng để chuyển các tập tin bằng cách sử dụng chuột thay vì ghi nhớ lệnh. FTP có thể cũng được sử dụng trực tiếp các ứng dụng khác để di chuyển các tập tin từ một nơi khác.

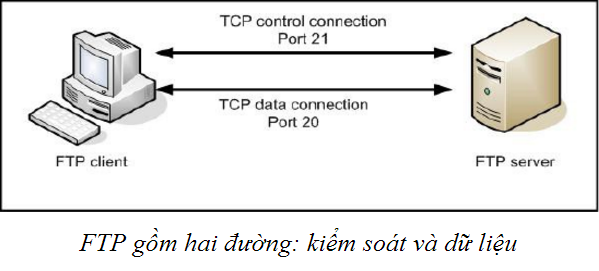
**2. Nguyên lý hoạt động**

**2.1. Mô hình hoạt động của FTP**

Không giống như các ứng dụng khác chạy trên nền TCP, FTP không chỉ cần một kết nối TCP mà các mô hình FTP được thiết kế xung quanh 2 kênh logic trong quá trình giao tiếp giữa client FTP và server FTP là control connection và data connection:

**Control connection:** Đây là kết nối TCP logic chính được tạo ra khi phiên làm việc được thiết lập. Nó được duy trì trong suốt phiên làm việc và chỉ cho các thông tin điều khiển đi qua ví dụ như lệnh và trả lời. Nó không được sử dụng để gửi dữ liệu.

**Data connection:** Mỗi khi dữ liệu được gửi từ sever tới client hoặc ngược lại, một kết nối dữ liệu TCP riêng biệt được thiết lập giữa chúng. Dữ liệu được truyền qua kết nối này. Khi hoàn tất việc truyền dữ liệu, kết nối được hủy bỏ.

Sự phân chia thành 2 kênh riêng biệt này tạo ra sự linh động trong cách hoạt động của FTP nhưng nó cũng tạo ra sự phức tạp cho FTP.

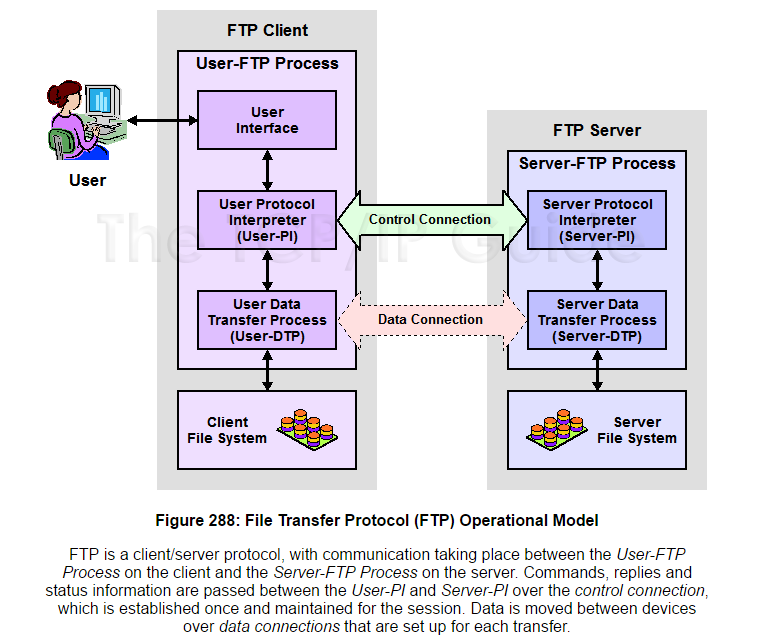
Do chức năng điều khiển và dữ liệu được truyền tải bằng cách sử dụng các kênh riêng biệt nên mô hình FTP chia phần mềm trên mỗi thiết bị thành 2 thành phần giao thức logic chịu trách nhiệm cho mỗi kênh:

**Thành phần protocol interpreter (PI):** Là thành phần quản lý kênh điều khiển, phát và nhận lệnh và trả lời.

**Thành phần data transfer process (DTP):** chịu trách nhiệm gửi và nhận dữ liệu giữa client và server.

Ngoài hai thành phần trên, tiến trình bên phía người dùng còn có 1 thành phần thứ ba là giao diện người dùng (user interface) dùng để tương tác với người dùng FTP, thành phần này không có ở phía sever.

Như vậy, có hai thành phần trong tiến trình bên phía server và ba thành phần trong tiến trình bên phía client. Các thành phần này được nhắc đến trong các mô hình FTP bằng cái tên cụ thể, được sử dụng trong các tiêu chuẩn để mô tả chi tiết hoạt động của giao thức. Dưới đây là mô hình hoạt động của FTP



**Tiến trình bên phía server:**

**Server Protocol Interpreter (Server-PI):** chịu trách nhiệm quản lý điều khiển kết nối trên server. Nó lắng nghe yêu cầu kết nối hướng từ users trên cổng dành riêng. Khi kết nối đã được thiết lập,nó nhận lện từ User-PI, gửi trả lời lại và quản lí tiến trình truyền dữ liệu trên server.

**Server Data Transfer 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 bên phía Client:**

**User Interface:** Đây là chương trình được chạy trên máy tính,nó cung cấp giao diện xử lí cho người dùng. Nó cho phép sử dụng các lệnh đơn giản hướng tới người dùng, và cho phép người điều khiển phiên FTP theo dõi được các thông tin và kết quả xảy ra trong tiến trình.

**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 hiện ra yêu cầu tới phía server-PI. Khi kết nối đã được thiết lập, nó xử lí 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.

**2.2. Thiết lập kênh điều khiển và chứng thực người dùng trong FTP**

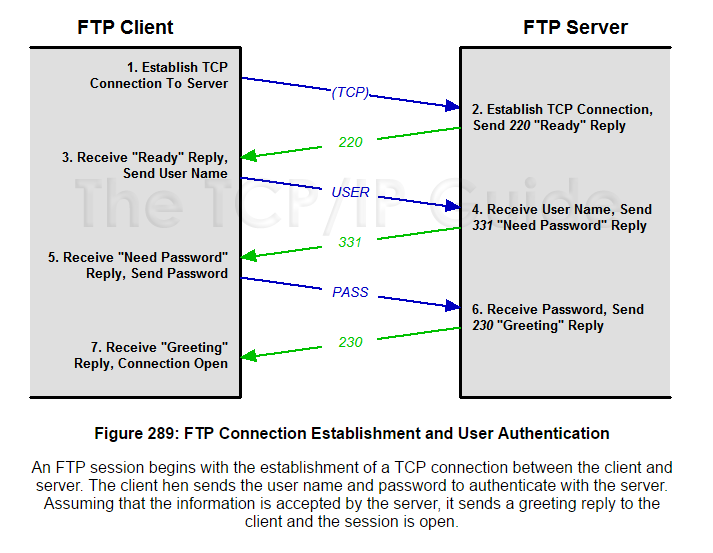
Trước khi kết nối được sử dụng để thực sự truyền file, kênh điều khiển cần phải được thiết lập. Như trong các giao thức client/server khác, FTP server tuân theo một luật passive trong kênh điều khiển. Bộ phận Server Protocol Interpreter (Server-PI) sẽ lắng nghe cổng TCP dành riêng cho kết nối FTP là 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 kì 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 thực hiện 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 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 FTP khá đơn giản, chỉ là cung cấp username/password để đăng nhập. Quy trình này được thể hiện trong hình sau: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 lời một thông báo tới người dùng rằng phiên làm việc đã được mở. Nếu người dùng không hợp lệ, server yêu cầu người dùng thực hiện lại việc chứng thực. Sau này 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 cụ thể dùng đọc và viết lên server, trong khi những người khác chỉ có thể lấy các tập tin. Người quản trị mạng có thể nhờ đó mà đáp ứng đúng các nhu cầu truy cập FTP.

Một khi kết nối 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 kì đầu của Internet. Ngày nay, nó không còn đảm bảo 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 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.

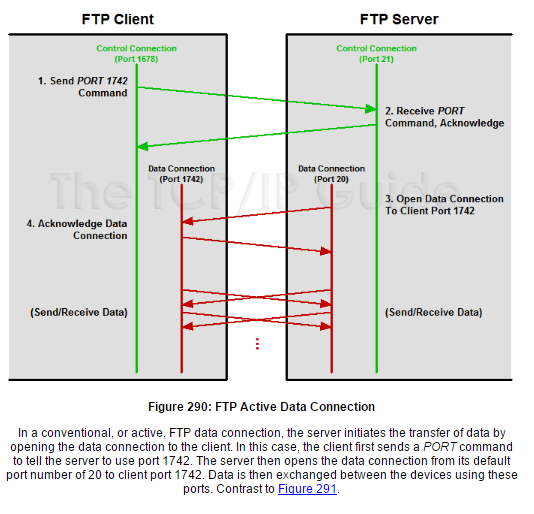
**Quản lí kênh dữ liệu FTP**

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, những dữ liệu thì không.

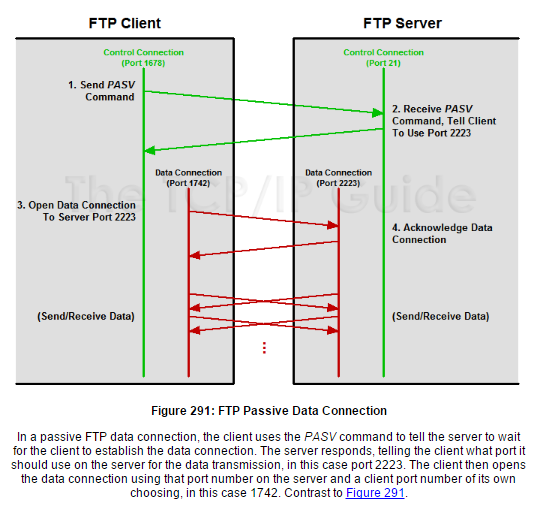
Mỗi khi cần phải truyền dữ liệu giữa các server và client, một kênh dữ liệu cần phải được tạo ra. Kênh dữ liệu kết nối bộ phận User-DTP và Server-DTP. Kết nối này cần thiết cho cả hoạt động truyền file trực tiếp (gửi hoặc nhận một file) cũng như đối với việc truyền dữ liệu ngầm, như là yêu cầu một danh 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 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. Normal (Active) Data Connections: Phương thức đầu tiên được gọi là tạo kết nối dữ liệu bình thường (vì đây là phương pháp mặc định) hay còn gọi là kết nối kênh dữ liệu ở dạng chủ động. Trong dạng kết nối này, phía Server-DTP tạo kênh dữ liệu bằng cách mở một cổng kết nối tới User-DTP. Server sử dụng cổng đặc biệt được dành riêng cho keetsnoois dữ liệu là cổng số 20. Trên máy client, cổng mặc định được sử dụng chính là cổng được sử dụng để kết nối điều khiển, nhưng server sẽ thường chọn mỗi cổng khác nhau cho mỗi chuyển giao.

Một ví dụ cụ thể cho cách làm việc này: Giả sử User-PI thiết đã thiết lập một kết nối điều khiển từ cổng bất kì của nó là cổng số 1678 tới cổng điều khiển FTP bên phía server là cổng số 21. Sau đó, để tạo một kết nối dữ liệu cho quá trình truyền dữ liệu, Server-PI sẽ báo cho phía Server-DTP để khởi tạo một kênh kết nối TCP từ cổng 20 tới cổng 1678 của phía client. Client sẽ nhận biết được điều này và sau đó dữ liệu được gửi đi theo cả hai chiều (nhớ rằng TCP truyền theo cả 2 chiều).

Trên thực tế, có kết nối điều khiển và kết nối bên phía Client trên cùng một cổng không là một ý tưởng hay, nó làm phức tạp các hoạt động của FTP và có thể dẫn tới một số vấn đề khó khan. Do đó, Client chỉ định một cổng khác bằng cách sử dụng lệnh PORT trước khi truyền dữ liệu.Ví dụ, giả sử phía client chỉ định cổng 1742 với lệnh PORT.Sau đó. phía Server-DTP tạo một kết nối từ cổng số 20 của nó tới cổng 1472 bên phía client thay cho 1678. Tiến trình này được trình bày trong hình sau:

**Passive Data Connections:** Phương pháp thứ hai được gọi là thiết lập kết nối dạng bị động. Các client sẽ gọi server là “bị động” (passive), điều này có nghĩa là nó chấp nhận một yêu cầu kết nối dữ liệu được khởi tạo bởi Client. Server trả lời lại phía client với địa chỉ IP cũng như địa chỉ cổng mà nó sẽ sử dụng. Sau đó phía Server-DTP lắng nghe trên cổng này một kết nối TCP đến từ User-DTP. Theo mặc định, phía client sẽ sử dụng cùng cổng mà nó sử dụng cho kết nối điều khiển như trong trường hợp chủ động. Tuy nhiên, trong phương pháp này, Client cũng có thể chọn sử dụng một cổng khác cho kết nối dữ liệu nếu cần thiết.

Xét lại ví dụ ở trên, với kết nối điều khiển từ cổng 1678 bên phía client tới cổng 21 bên phía server, nhưng lần này ta xem xét dữ liệu được chuyển đi bằng phương pháp kết nối dạng bị động. Phía client sẽ hiện lệnh PASV để nói với server rằng nó muốn sử dụng điều khiển dữ liệu dạng bị động. Server-PI sẽ trả lời lại với một giá trị cổng mà phía client đã sử dụng, cổng 2223. Server-PI sau đó sẽ hướng dẫn Server-DTP lắng nghe trên cổng 2223 này. User-PI sẽ hướng dẫn User-DTP tạo ra một kết nối từ cổng 1742 bên phía client tới cổng 2223 bên phía server. Phía server sẽ nhận biết được điều này và sau đó dữ liệu sẽ được gửi và nhận theo cả hai hướng. Quá trình này diễn ra trong mô hình sau:

**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. FTP có ba phương thức truyền dữ liệu, 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 đầy 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 cầu 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 thưc, 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 nến 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.

**Đặc trưng của dữ liệu trong FTP**

Cách xây dựng ổng quát giao thức truyền tập tin đã có để làm cho nó xử lý tất cả các tập tin như là "hộp đen". Các tập tin sẽ được coi như là một tập hợp các byte. FTP sẽ không chú ý đến nội dung của các tập tin, và sẽ chỉ đơn giản là di chuyển các tập tin, một byte cùng một thời điểm, từ nơi này sang nơi khác. Ở đây, nó tương tự như là "sao chép" lệnh được thực hiện trên hầu hết các hệ thống tập tin, tương tự như vậy có thể tạo ra một bản sao mà không cần tìm kiếm vào tập tin để xem những gì nó có.

Vì vậy những gì sẽ là các vấn đề với điều đó? Đối với một số loại tập tin, sẽ không có vấn đề gì, nhưng đối với những loại khác, nó là cả một vấn đề. Lý do là một số loại tập tin khác nhau sử dụng trên hệ thống khác nhau. Nếu bạn sao chép một tập tin từ một nơi khác trên cùng một máy tính bằng cách sử dụng lệnh "copy", không có vấn đề gì vì cùng đại diện cho các tập tin được sử dụng ở khắp mọi nơi trong máy tính đó. Nhưng khi bạn sao chép nó vào một máy tính sử dụng một đại diện khác nhau, bạn có thể gặp khó khăn.

Ví dụ phổ biến nhất của điều này là một loại tập tin có thể làm bạn ngạc nhiên: các tập tin văn bản đơn giản. Tất cả các tập tin văn bản ASCII sử dụng bộ ký tự ASCII, nhưng chúng khác nhau ở các kí tự điều khiển được sử dụng để đánh dấu sự kết thúc của một dòng văn bản. Trên UNIX, một dòng nguồn cấp dữ liệu ký tự (LF) được sử dụng; trên các máy tính Apple, một vận chuyển trở lại (CR); và máy windownsử dụng cả hai (CR + LF).

Nếu bạn di chuyển một tập tin văn bản từ một trong những loại hệ thống khác bằng cách sử dụng FTP thông thường, tất cả dữ liệu sẽ được được di chuyển chính xác. Di chuyển một tập tin văn bản từ một hệ thống UNIX tới PC cũng như là chuyển một tập hợp các byte, có nghĩa là chương trình sẽ không nhận ra đúng kết thúc của các dòng dấu hiệu. Tránh tình trạng khó khăn này đòi hỏi phải dịch FTP chuyển qua ý tưởng rằng “all files are just bytes” và kết hợp một chút thông minh để xử lý các loại khác nhau của tập tin. Các tiêu chuẩn FTP công nhận điều này bằng cách cho phép đặc tảc một số chi tiết nhất định về biểu diễn nội dung tập tin trước khi nó được chuyển.

**FTP Data Types:** Phần đầu tiên của thông tin có thể được đưa ra về tập tin là kiểu dữ liệu của nó, cái mà có thể dứng ra đại diện tổng thể cho tập tin. Có bốn kiểu dữ liệu khác nhau được quy định trong chuẩn của FTP.

* ASCII: Định nghĩa một file văn bản ASCII, với dòng được đánh dấu bởi một số loại dòng đánh dấu kết thúc như mô tả ở trên.
* EBCDIC: Khái niệm tương tự như loại ASCII, nhưng đối với tập tin bằng cách sử dụng kí tự EBCDIC của IBM đặt.
* Image: Các tập tin đã không có cấu trúc nội bộ chính thức và được gửi một byte tại một thời điểm mà không có bất kỳ xử lý; đây là chế độ "hộp đen" đã đã đề cập ở trên.
* Local: Kiểu dữ liệu này được sử dụng để xử lý các tập tin có thể lưu trữ dữ liệu trong byte logic có chứa một số bit khác 8. Cách xác định loại này cùng với cách dữ liệu có cấu trúc cho phép dữ liệu được lưu trữ trên hệ thống đích một cách phù hợp với đại diện địa phương của nó.

Trong thực tế, hai loại kiểu dữ liệu thường xuyên nhất được sử dụng là ASCII và Image. Kiểu ASCII được sử dụng cho các tập tin văn bản, và cho phép chúng được di chuyển giữa các hệ thống với dòng kết thúc mã chuyển đổi tự động. Loại hình ảnh được sử dụng cho các tập tin nhị phân chung, chẳng hạn như đồ họa hình ảnh, tập tin ZIP và các dữ liệu khác được thể hiện một cách phổ quát. Nó cũng thường được gọi là nhị phân loại vì lý do đó.

**FTP Format Control:** Đối với loại ASCII và EBCDIC, FTP định nghĩa một tham số tùy chọn được gọi là format contrl. Điều này cho phép người dùng chỉ định một đại diện cụ thể cho các định được sử dụng để mô tả một tập tin theo chiều dọc dạng như thế nào. Ba tùy chọn là:

* Không in: Mặc định, cho biết không có định dạng thẳng đứng
* Dạng thức Telnet: Chỉ ra rằng định dạng thẳng đứng điều khiển nhân vật, như được chỉ định trong giao thức Telnet, được sử dụng trong tập tin này.
* Kiểm soát vận chuyển / FORTRAN: tệp sử dụng định dạng điều khiển ký tự làm ký tự đầu tiên của mỗi dòng, như quy định đối với FORTRAN ngôn ngữ lập trình.

Các tùy chọn kiểm soát định dạng được tạo ra cho mục đích cụ thể đúng cách xử lý các tập tin chuyển giao từ các thiết bị máy chủ đến máy in. Nó không được sử dụng ngày nay, hoặc nếu nó được sử dụng, nó chỉ là trong ứng dụng đặc biệt.

**FTP Data Structures:** Ngoài ra để xác định kiểu dữ liệu của tập tin, nó cũng có thể để xác định cấu trúc dữ liệu của tập tin. Hiện có 3 khả năng:

* Cấu trúc tập tin: Các tệp là một dòng byte với cấu trúc bên trong không tiếp giáp.
* Cấu trúc bản ghi: Tập tin bao gồm một bộ hồ sơ tuần tự, mỗi trong số đó được phân cách bởi một điểm đánh dấu kết thúc của hồ sơ.
* Cấu trúc trang: Các tập tin chứa một tập hợp các dữ liệu đặc biệt được lập chỉ mục trang. Cấu trúc tập tin mặc định và được sử dụng cho hầu hết các loại tập tin. Cấu trúc bản ghi có thể được sử dụng cho các tập tin văn bản ASCII, nhưng chúng thường được gửi với các cấu trúc file thường xuyên sử dụng kiểu dữ liệu ASCII. Cấu trúc trang không được sử dụng phổ biến; ban đầu được tạo cho một loại máy tính được sử dụng trong thời kì đầu ARPANET.