**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**



**Báo cáo môn mạng máy tính**

**ĐỀ TÀI 3: Tìm hiểu hoạt động và cài đặt dịch vụ FTP**

**Giảng viên hướng dẫn : Bùi Trọng Tùng**

**Sinh viên thực hiện :** Nguyễn Bình Minh – 20152453

Vũ Minh Quân – 20153054

Vũ Đức Hiệp – 20151447

**Phần 1: Giới Thiệu**

- Giao thức FTP (File Transfer Protocol) thường được sử dụng để trao đổi tập tin qua mạng dùng giao thức TCP/IP.

- Mục đích của FTP như đã phác thảo trong RFC:

+Khuyến khích việc dùng chung tệp tin.

+Khuyến khích việc sử dụng máy tính ở xa một cách gián tiếp.

+Che đậy sự khác biệt về hệ thống lưu trữ tập tin giữa các máy chủ.

+Truyền tải dữ liệu một cách đáng tin cậy và có hiệu quả cao.

- Hiện nay có rất nhiều server FTP như : Filezilla, Cute FTP, Flash FXP, FTPNet, smartFTP… Trong các Server này nhóm em quyết định chọn Filezilla là FTP Server bởi vì đây là phần mềm nguồn mở.

**Phần 2: Lịch sử giao thức**

- FTP đã có quá trình phát triển dài qua nhiều năm. Phụ lục III là một tài liệu được biên soạn dần theo thời gian về [FTP.](ftp://FTP.Chúng) Nó bao gồm những kỹ thuật được đề xuất đầu tiên để chuyển file, chúng được phát triển để thực hiện trên các máy chủ tại M.I.T (RFC 114), vào năm 1979, cộng với những nhận xét và thảo luận ở RFC 141.

- RFC 172 cung cấp giao thức hướng người dùng cho việc truyền tệp giữa 2 máy tính. Sửa đổi ở RFC 265, xem lại FTP để xem xét bổ xung khi RFC 281 gợi ý những thay đổi tiếp theo. Việc sử dụng 1 “set data type” (kiểu dữ liệu tập hợp) giao dịch đã được đề xuất trong RFC 294 vào tháng 1 năm 1982.

- RFC 354 không dùng RFC 264 và 265. Giao thức truyền file bây giờ được định nghĩa là một giao thức truyền tệp tin giữa các host trên Arpanet, với chức năng chính là chuyển các tệp tin hiệu quả, đáng tin cậy giữa các host và cho phép lưu trữ file từ xa.

-RFC 385 nhận xét thêm về các lỗi, điểm nhấn, thêm vào các giao thức. RFC cung cấp báo cáo tình trạng trên máy chủ làm việc và người dùng FTP. RFC 430 thêm nhận xét trên FTP. Cuối cùng 1 tài liệu chính thức được xuất bản dưới dạng là RFC 454.

-Tháng 7 năm 1973 những thay đổi đáng kể từ những phiên bản cuối cùng của FTP đã được thực hiện, nhưng cấu trúc chung vẫn được giữ nguyên. RFC 542 được đề xuất như bản đặc tả chính thức. Rồi các bản đề xuất liên tiếp RFC 607, 614, 624, 686, 691 đã tiếp tục thêm nhận xét thay đổi thiết kế và sửa đổi nhỏ.Và sau khi thêm lệnh tùy chọn, sửa và cải thiện tính năng giao thức RFC 959 được xem là tài liệu chính thức.

**Phần 3 ứng dụng của giao thức**

- Sử dụng trình FTP để đồng bộ tệp tin hằng ngày (như ứng dụng quản lý file)

- Cho phép chuyển tệp tin đến các thiết bị kết nối trong mạng

- Toàn bộ file được truyền từ thiết bị này đến thiết bị khác và được copy vào bộ nhớ, còn HTTP chỉ truyền tải nội dung của 1 trang web vào 1 trình duyệt để xem

- FTP là 1 hệ thống 2 chiều (two-way system) khi các file được truyền qua lại giữa máy chủ và máy trạm. Nên nó được dùng phổ biến để upload các trang web từ nhà thiết kế web lên 1 máy chủ host trên internet.

**Phần 4 Một số thuật ngữ**

**-** User Interface: Cung cấp giao diện người dùng và điều khiển phần phiên dịch giao thức phía Client.

- Control connection : Kênh điều khiển, đường kết nối giữa USER-PI và SERVER-PI trao đổi các lệnh và phản hồi. Kết nối này dựa theo giao thức Telnet.

- Data connection: Kênh dữ liệu. Có thể thiết lập giữa 1 server-DTP và user-DTP hoặc giữa hai server-DTP. Dữ liệu được truyền có thể là 1 phần của một tệp tin hoặc một số tệp tin.

-DTP: The Data Tranfer Process thiết lập và quản lý kết nối dữ liệu. DTP có thể ở chế độ chủ động hoặc bị động.

-EOR: end-of-record điều kiện xác định một bản ghi đang được chuyển giao

-Error recovery: thủ tục cho phép người sử dụng phục hồi từ các lỗi nhất định chẳng hạn như sự thất bại của quá trình chuyển file.

-FTP command: một tập các lệnh cho phép để điều khiển thông tin trôi chảy từ user-FTP đến tiến trình server-FTP

-PI: the Protocol Interpreter. Các bên user và server của giao thức có vai trò thực hiện khác nhau trong uesr-PI và server-PI.

**Phần 5 Sơ lược về giao thức TCP**

- TCP/IP là bộ giao thức cho phép kết nối các hệ thống mạng không đồng nhất với nhau. Mô hình TCP/IP gồm 5 tầng: Application, Transport, Network, Datalink, Physical

- TCP (Transmission Control Protocol) là một trong các giao thức tầng giao vận. TCP cung cấp luồng dữ liệu tin cậy giữa hai máy trạm, sử dụng cơ chế như chia nhỏ gói tin với kích thước hợp lí cho tầng mạng bên dưới, báo nhận gói tin (ACK), đặt thời gian (timeout).. Để cho dữ liệu đến đích đúng thứ tự, gói tin thất lạc thì được gửi lại, lặp gói tin sẽ được loại bỏ, kiểm soát tắc nghẽn.

- TCP đòi hỏi thiết lập trước khi truyền dữ liệu, quá trình bắt tay 3 bước (3 way handshake).

- Hỗ trợ nhiều giao thức trên Internet trong đó có FTP.

**Phần 6 Câu lệnh**

**6.1 Khuôn dạng câu lệnh**

6.1.1 Lệnh điều khiển truy nhập

- USER NAME (USER) : trường tham số là 1 chuỗi Telnet xác định người dùng . Việc nhận dạng người dùng được yêu cầu bởi Server để truy cập vào hệ thống tệp tin của nó. Lệnh này thường là lệnh đầu tiên được truyền bởi người dùng sau các kết nối điều khiển được thiết lập. Thông tin nhận dạng bổ sung dưới hình thức lệnh mật khẩu (password). Server có thể yêu cầu lệnh USER mới vào bất kì thời điểm nào để thay đổi điều khiển truy nhập hoặc thông tin tài khoản.

USER <SP> <username> <CRLF>

- PASSWORD (PASS) : trường tham số là 1 chuỗi Telnet chỉ rõ password của người dùng. Từ khi đến phần thông tin mật khẩu là khá nhạy cảm, nó phải được che giấu kín. Server không có cách nào để đạt được điều này.Do đó nó là trách nhiệm của người sử dụng FTP để mà ẩn thông tin mật khẩu nhạy cảm.

PASS <SP> <password> <CRLF>

- ACCOUNT (ACCT) : trường tham số là 1 chuỗi Telnet xác định tài khoản người dùng. Lệnh này không nhất thiết liên quan đến lệnh USER, vài nơi yêu cầu 1 tài khoản login, hoặc chỉ truy cập cụ thể chẳng hạn như lưu trữ file. Trong trường hợp thứ hai lệnh có thể đến bất cứ lúc nào.

Mã trả lời phân biệt các trường hợp này tự động: khi thông tin tài khoản yêu cầu để đăng nhập, phản hồi thành công yêu cầu PASSword là mã 331. Mặt khác nếu thông tin tài khoản không yêu cầu đăng nhập trả lời một lệnh PASSword thành công là 230

ACCT <SP> <account-information> <CRLF>

- CHANGE WORKING DIRECTORY (CWD) : Lệnh này cho phép người dùng làm việc với các thư mục khác nhau hoặc tập dữ liệu cho việc lưu file hoặc lấy về mà không thay đổi thông tin tài khoản. Tham số là một tên dường dẫn xác định một thư lục hoặc hệ thống phụ thuộc tệp tin định danh nhóm.

CWD <SP> <pathname> <CRLF>

- CHANGE TO PARENT DIRECTORY (CDUP) : Lệnh này là trường hợp đặc biệt của CWD, bao gồm đơn giản hóa việc thực hiện chuyển cây thư mục giữa các hệ điều hành có cú pháp khác nhau để đặt tên thư mục gốc.Mã trả lời giống mã trả lời CWD

CDUP <CRLF>

- STRUCTURE MOUNT (SMNT) : Lệnh này cho phép cho phép người dùng mount một cấu trúc dữ liệu hệ thống tệp tin khác mà không thay đổi đăng nhập của mình. Tham số là một tên dường dẫn xác định một thư lục hoặc hệ thống phụ thuộc tệp tin định danh nhóm.

SMNT <SP> <pathname> <CRLF>

- LOGOUT (QUIT) : Lệnh kết thúc 1 user và nếu không phải đang tiến hành truyền file, server sẽ đóng kết nối điều khiển. Nếu đang truyền file thì kết nối tiếp tục được mở để phản hồi kết quả và server sau đó sẽ đóng nó. Nếu người dùng đang truyền tệp tin cho một số user nhưng không muốn đóng kết nối và sau đó mở lại kết nối cho từng user thì lệnh REIN nên được sử dụng thay vì QUIT.

Đóng kết nối điều khiển 1 cách bất ngờ sẽ gây ra server ảnh hưởng hành động của abort (ABOR) và logout (QUIT)

QUIT <CRLF>

6.1.2 Lệnh truyền tham số

- DATA PORT (PORT) : tham số đặc tả HOST PORT cổng dữ liệu được sử dụng để kết nối dữ liệu. Mặc định cho cả cổng dữ liệu user và server, dưới mức bình thường lệnh và mã trả lời là không cần thiết. Nếu lệnh được sử dụng thì tham số là sự nối tiếp của 32 bits internet host address và 16 bits địa chỉ cổng TCP. Thông tin địa chỉ được chia thành các trường 8 bits và giá trị của từng trường được truyền như số thập phân. Các trường phân tách bởi dấu “,” : PORT h1,h2,h3,h4,p1,p2 (h1 là giá trị biểu diễn 8 bits cao của internet host address)

Host-port gồm host-number và port-number

PORT <SP> <host-port> <CRLF>

- PASSIVE (PASV) : Lệnh này yêu cầu server-DTP “listen” trên cổng dữ liệu (không phải là cổng dữ liệu mặc định) và đợi 1 kết nối thay vì nhận 1 lệnh truyền

PASV <CRLF>

- REPRESENTATION TYPE (TYPE) : dùng để chuyển đổi kiểu file sẽ được truyền. Tham số xác định representation type như miêu tả trong phần biểu diễn và lưu trữ dữ liệu. Một số loại có tham số thứ hai, tham số đầu được kí hiệu bởi 1 kí tự Telnet đơn, tham số thứ hai định dạng cho ASCII và EBCDIC ( là số nguyên thập phân để chỉ bytesize) .

Mã sau được chỉ định cho type:

A – ASCII | N – Non-print

E – EBCDIC | T – Telnet format effector

| C – Carriage Control

I – Image

L – local byte size

Kiểu biểu diễn mặc định là ASCII non-print . Nếu định dạng tham số thay đổi và sau đó chỉ thay đổi tham số đầu, thì định dạng sẽ trả về non-print mặc định.

TYPE <SP> <type-code> <CRLF>

- FILE STRUCTURE (STRU) : tham số là một mã kí tự Telnet đơn xác định cấu trúc file được mô tả trong phần biểu diễn và lưu trữ dữ liệu. Các mã sau chỉ định cấu trúc:

F – File

R – Record Structure (cấu trúc bản ghi)

P – Page Structure

Cấu trúc mặc định là File

STRU <SP> <structure-code> <CRLF>

- TRANSFER MODE (MODE) : tham số là một mã kí tự Telnet đơn xác định chế độ truyền dữ liệu được mô tả trong phần biểu diễn và lưu trữ dữ liệu. Các mã sau chỉ định chế độ truyền:

S – Stream

B – Block

C – Compressed

Chế độ truyền mặc định là Stream

MODE <SP> <mode-code> <CRLF>

6.1.3 Lệnh dịch vụ FTP

- RETRIEVE (RETR) : Lệnh này làm server-DTP truyền bản copy file đến đường dẫn cụ thể, đầu kia của kết nối dữ liệu có thể là server-DTP hoặc user-DTP. Trạng thái và nội dung của file trên phía máy chủ sẽ không bị ảnh hưởng

RETR <SP> <pathname> <CRLF>

- STORE (STOR) : lệnh này làm server-DTP chấp nhận truyền dữ liệu qua kết nối dữ liệu và lưu dữ liệu như file phía server. Nếu file chỉ định trong đường dẫn đã tồn tại phía máy chủ, thì nội dung của nó sẽ được thay thế bởi dữ liệu đang chuyển giao. Ngược lại, sẽ tạo ra file mới.

STOR <SP> <pathname> <CRLF>

-STORE UNIQUE (STOU) : Lệnh này như lệnh STOR ngoại trừ tệp tin được tạo trong thư mục hiện hành dưới tên duy nhất. Mã phản hồi 250 phải bao gồm tên được tạo

STOU <CRLF>

- APPEND (APPE) : lệnh này làm server-DTP chấp nhận dữ liệu truyền qua kết nối dữ liệu và lưu dữ liệu ở một file phía server. Nếu file chỉ định trong đường dẫn đã tồn tại ở phía máy chủ, thì dữ liệu sẽ được viết thêm vào file đó. Ngược lại, sẽ tạo file mới.

APPE <SP> <pathname> <CRLF>

- ALLOCATE (ALLO) : lệnh này yêu cầu server dự trữ đủ dung lượng để chứa file mới được chuyển.Tham số là số nguyên thập phân biểu diễn số byte. Tiếp sau lệnh này sẽ là STOR hoặc APPE.

ALLO <SP> <decimal-integer>

- RESTART (REST) : trường tham số thể hiện server marker tại đó việc truyền file sẽ được khởi động lại

REST <SP> <marker> <CRLF>

- RENAME FROM (RNFR) : lệnh chỉ rõ đường dẫn cũ của file đổi tên

RNFR <SP> <pathname> <CRLF>

- RENAME TO (RNTO) : lệnh chỉ ra đường dẫn mới của file, kết hợp sau lệnh RNFR sẽ đổi tên file

RNTO <SP> <pathname> <CRLF>

- ABORT (ABOR) : lệnh yêu cầu server hủy bỏ lệnh FTP service trước và bất kì liên kết truyền dữ liệu.

ABOR <CRLF>

- DELETE (DELE) : lệnh này làm cho file chỉ định trong đường dẫn sẽ được xóa ở phía server

DELE <SP> <pathname> <CRLF>

- REMOVE DIRECTORY (RMD) : lệnh này sẽ xóa thư mục chỉ định trong đường dẫn

RMD <SP> <pathname> <CRLF>

- PRINT WORKING DIRECTORY (PWD) : lệnh yêu cầu trả về tên thư mục làm việc hiện tại

PWD <CRLF>

- LIST (LIST) : lệnh làm danh sách được gửi từ server đến passive DTP. Nếu đường dẫn cụ thể là 1 thư mục hoặc nhóm file, server sẽ truyền danh sách file trong thư mục đó.

LIST [<SP> <pathname>] <CRLF>

- NAME LIST (NLST) : lệnh làm danh sách thư mục được gửi từ server đến phía user.

NLST [<SP> <pathname>] <CRLF>

- SITE PARAMETERS (SITE) : được sử dụng bởi server cung cấp dịch vụ của hệ thống nhưng không đầy đủ bao gồm lệnh ở trong giao thức

SITE <SP> <string> <CRLF>

- SYSTEM (SYST) : lệnh tìm kiểu hê điều hành tại server.

SYST <CRLF>

- STATUS (STAT) : lệnh sẽ làm trả lời trạng thái gửi qua kết nối điều khiển dưới hình thức trả lời. Nếu có đối số lệnh này tương tự lệnh LIST ngoại trừ được truyen qua kết nối điều khiển.

STAT [<SP> <pathname>] <CRLF>

- HELP (HELP) : lệnh này yêu cầu máy chủ gửi thông tin hữu ích về tình trạng thực hiện qua kết nối điều khiển.

HELP [<SP> <string>] <CRLF>

- NOOP (NOOP): lệnh này không làm gì, ngoài việc server gửi OK reply

NOOP <CRLF>

**6.2 Khuôn dạng thông điệp trả lời**

Trả lời lệnh FTP được tạo ra để đảm bảo đồng bộ trong requests và action trong quá trình truyền file, đảm bảo tiến trình user luôn biết trạng thái của server. Mỗi lệnh phải tạo ra ít nhất 1 reply (trả lời) . Ngoài ra một số lệnh xuất hiện trong các nhóm tuần tự, như USER, PASS, ACCT hoặc RNFR, RNTO

FTP reply bao gồm 3 chứ số sau là vài đoạn xâu và chấm dứt bởi mã Telnet end-of-line. Định dạng reply gồm nhiều dòng, dòng đầu sẽ bắt đầu với mã trả lời yêu cầu tiếp theo là dấu gạch nối “-“ và đoạn text . Dòng cuối sẽ bắt đầu với mã giống mã đầu tiếp theo là Space <SP> , tùy chọn một số text , và cuối cùng mã Telnet end-of-line.

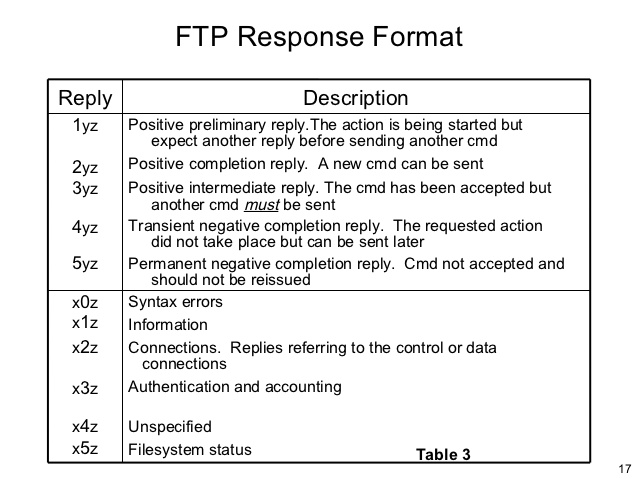
Example: 123-First line

Second line

234 A line beginning with numbers

123 The last line

3 số của mỗi reply có ý nghĩa đặc biệt. Chữ số đầu biểu thị response tốt , xấu hoặc không đầy đủ, không phức tạp tiến trình user sẽ xác định hành động tiếp theo. Chữ số thứ hai cho biết loại lỗi xảy ra. Chứ số thứ ba dành cho thứ tự hay phân cấp của thông tin ( e.g RNTO mà không RNFR trước )



1yz Trả lời chấp nhận sơ bộ. Đợi một trả lời tiếp theo trước khi gửi lệnh khác

2yz Trả lời chấp nhận hoàn tất. lệnh cuối cùng đã thực hiện thành công.

3yz Trả lời chấp nhận tạm thời. Một lệnh nữa phải được gửi ( cần thêm thông tin từ phía client)

4yz Trả lời hoàn tất phủ nhận nhất thời. Hoạt động yêu cầu không được thực hiện nhưng có thể thử lại

5yz Trả lời hoàn tất phủ nhận lâu dài. Hoạt động yêu cầu không được thực hiện và không nên thử lại

------------------------------------------------------------------------------------------------------

x0z lỗi cú pháp

x1z yêu cầu thông tin, như status hoặc help

x2z trả lời đề cập đến kết nối dữ liệu và điều khiển

x3z trả lời cho quá trình đăng nhập và accounting

x4z không xác định

x5z trạng thái hệ thống file phía máy chủ với yêu cầu nào đó

Chữ số thứ 3 cho phép phân loại tốt hơn với mỗi ý nghĩa của chữ số thứ 2. Triển khai server không nên phát minh ra mã mới mà nên thích ứng với mã đã được xác định

Ngoài ra còn một số FTP reply khác từ các rfc sau như: RFC 2228 bổ sung Protected replies (600,631,632,633) , 10000 Common Winsock Error Codes,…

**6.3 Một số câu lệnh ví dụ**

- Một số lệnh có tham số thường dùng:

HELP : yêu cầu trợ giúp

USER minh : đăng nhập với tên “minh”

PASS 123 : chỉ ra rằng mật khẩu là “123”

CWD /FTP/test : chuyển sang thư mục /FTP/test

MKD test : yêu cầu tạo thư mục tên “test”

RMD test : xóa thư mục tên test

MODE S : thiết lập chế độ truyền là Stream

TYPE I : sẽ dùng kiểu dữ liệu nhị phân

RETR test.txt : yêu cầu download file tên “test.txt” trên thư mục hiện hanh

STOR test.txt : yêu cầu upload file “test.txt” lên

RNFR test.txt : yêu cầu đổi tên file test.txt

RNTO TEST.txt : đổi tên file test.txt thành TEST.txt

DELE TEST.txt : xóa file tên TEST.txt trên server

- Một số mã trả lời của FTP server:

- 200: lệnh được chấp nhận

- 500 : lỗi cú pháp không nhận thấy được lệnh

- 501: lỗi cú pháp trong tham số hoặc đối số

- 202 : lệnh không được thi hành

- 504 : tham số của lênh không được thực thi

- 211 : cho biết trạng thái của hệ thống

- 212 : cho biết trạng thái thư mục hiện thời

- 213 : cho biết trạng thái file

- 214 : gửi 1 thông điệp giúp đỡ

- 120 : sẵng sàng phục vụ

- 220 : dịch vụ sẵng sàng cho người dùng mới

- 221 : dịch vụ đóng kết nối điều khiển

- 421 : dịch vụ không hợp lý, đóng kết nối điều khiển

- 125 : kết nối dữ liệu đã được mở, đang bắt đầu trao đổi

- 225 : kết nối dữ liệu đã mở, nhưng không có tiến trình trao đổi

- 425 : không thể mở kết nối dữ liệu

- 226 : đóng kết nối dữ liệu

- 426 : đóng kết nối và hủy bỏ trao đổi

- 230 : người dùng đã đăng nhập thành công

- 530 : không đăng nhập được

- 331 : tên người dùng đúng và cần mật khẩu

- 332 : cần tài khoản để đăng nhập

- 532 : cần tài khoản để lưu trữ file

- 250 : yêu cầu file được chấp nhận

- 257 : tên đường dẫn đã được tạo

- 350 : yêu cầu file đang chờ thêm thông tin

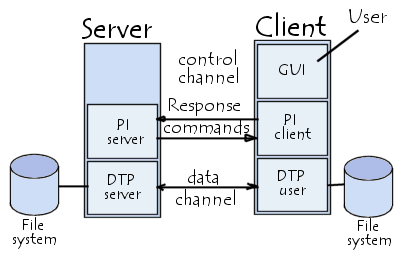
- 450 : yêu cầu file không được thực thi

- Một ví dụ khi thiết lập kết nối:

|  |  |
| --- | --- |
| Lệnh | Mã trả lời |
| Connected on port 21 | 220 Filezilla Server 0.9.60 beta |
| USER minh | 331 Password required for minh |
| PASS \*\*\* | 230 Logged on |
| SYST | 215 UNIX emulated by Filezilla |
| FEAT | 221-Features: MDTM  REST STREAM  …  221 End |
| PWD | 257 “/” is current directory |
| TYPE | 200 Type set to I |
| PASV | 227 Entering Passive Mode (127.0.0.1.255.9) |
| MLSD | 150 Opening data chennel for directory listing of “/” |
|  | 226 Successfully trasferred “/” |

**Phần 7 Nguyên lý hoạt động**

**7.1 Mô hình hoạt động**

****

Mô hình trên có thể là sơ đồ của dịch vụ FTP

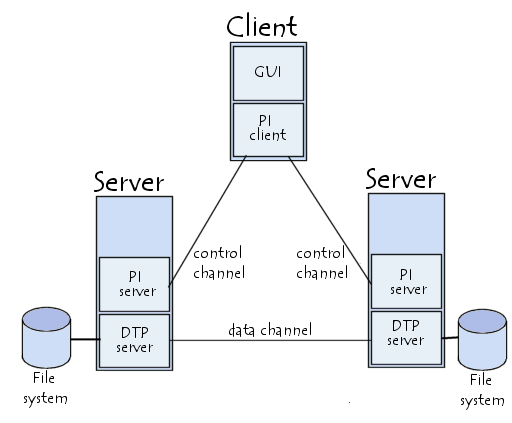
Note: +data connection có thể sử dụng ở cả hai hướng

+data connection không cần tồn tại tất cả mọi lúc

- User-PI bắt đầu kết nối điều khiển , control connection theo giao thức telnet. Khi khởi tạo kết nối của người sử dụng, commands standard FTP được phát ra bởi user-PI và truyền đến server sử lý thông qua kết nối điều khiển. Trả lời được gửi từ server-PI to user-PI qua kết nối điều khiển

- FTP commands chỉ rõ thông số của kết nối dữ liệu ( data port, tranfer mode, representation type, và structure) và loại của file hệ điều hành( store, retrieve, append, delete,..) . Nên chú ý là cổng dữ liệu không cần giống host bắt đầu lênh FTP thông qua kết nối điều khiển, nhưng user phải nghe trên cổng dữ liệu cụ thể. Kết nối dữ liệu thường dùng đồng thời cho cả gửi và nhận dữ liệu.

- ở trong một số tình huống user có thể yêu cầu truyền dữ liệu giữa 2 hosts. User cài đặt kết nối điều khiển giữa 2 server và sau đó chuẩn bị cho kết nối dữ liệu giữa chúng. Theo cách đó thông tin điều khiển chuyển cho user-PI còn còn dữ liệu truyền giữa 2 server. ở dưới là mô hình tác động qua lại giữa 2 server:



- Giao thức yêu cầu kênh kết nối điều khiển phải mở khi tiến hành truyền dữ liệu. Trách nhiệm của user để yêu cầu đóng kết nối điều khiển khi kết thúc sử dụng dịch vụ FTP. Server có thể hủy bỏ truyền dữ liệu nếu kết nối điều khiển bị đóng mà không có lệnh.

-Giao tiếp theo mô hình khách chủ (client/server) mỗi dịch vụ có cách gửi nhận dữ liệu theo quy ước riêng. TCP và UDP chỉ chịu trách nhiện đưa dữ liệu từ một máy tính này đến máy tính khác, còn dữ liệu còn dữ liệu đó sẽ gửi cho dịch vụ nào thì phải thông qua một quy định nữa là cổng(Port).

-Trước khi yêu cầu một dịch vụ trên máy chủ thực hiện điều gì đó máy khách phải có khả năng kết nối với máy chủ . Quá trình này được thực hiện thông qua cơ chế trìu tượng gọi là socket. Việc kết nối theo cơ chế Socket cần biết 2 thông tin chủ yếu đó là địa chỉ của máy cần kết nối và số hiệu cổng của chương trình dịch vụ.

**-**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.

-Bộ phần mềm FTP được cài đặt trên một thiết bị được gọi là một tiến trình:

+Tiến trình Server-FTP

+Tiến trình User-FTP

-Mô hình FTP chia quá trình truyền thông giữa bộ phận Server với bộ phân Client ra làm 2 kênh logic:

+Kênh điều khiển (control connection)

+Kênh dữ liệu (data connection)

- Kết nối điều khiển được dùng để truyền lệnh. Một vài lệnh có liên quan đến truyền dữ liệu giữa 2 hosts. Lênh truyền dữ liệu bao gồm MODE lệnh cụ thể để làm sao bit dữ liệu truyền được, và STRUcture và TYPE lệnh dùng xác định cách dữ liệu được biểu diễn. Truyền và biểu thị dữ liệu về cơ bản là độc lập nhưng chế độ truyền “stream” phụ thuộc vào thuộc tính cấu trúc file và nếu chế độ file nén được sử dụng bản chất byte phụ thuộc kiểu biểu diễn.

**7.2 Thiết lập kênh điều khiển, kết nối tới Server và đăng nhập**

**//….**

**7.3.1.Trình diễn và lưu trữ dữ liệu**

**-** Dữ liệu được truyền từ thiết bị lưu trữ trong sending host đến thiết bị lưu trữ trong receiving host. Thường cần thiết để biểu diễn chắc chắn chuyển đổi dữ liệu bởi vì biểu diễn dữ liệu ở 2 hệ thống khác nhau. Ví dụ NTV-ASCII trên các hệ thống khác nhau được biểu diễn bằng số bit khác nhau. Một vấn đề biểu diễn khác phát sinh khi truyền binary data( not character code) giữa 2 host với độ dài từ khác nhau. User nên lựa chọn chức năng thực hiện truyền và biểu diễn dữ liệu. Cần lưu ý rằng FTP cung cấp chỉ giới hạn kiểu dữ liệu biểu diễn. Muốn thực hiện những chuyển đổi vượt ra ngoài giới hạn này cần phải thực hiện trực tiếp bởi người sử dụng

7.3.1.1. kiểu dữ liệu (DATA TYPES)

- Biểu diễn dữ liệu được xử lý trong FTP bởi kiểu biểu diễn với user cụ thể. Loại biểu diễn này có thể ngầm định (ASCII hay EBCDIC) hoặc rõ ràng ( như local byte) xác định kích thước byte cho sự biểu diễn, được gọi là “logical byte size”. Lưu ý rằng không giống với kích thước byte được sử dụng để truyền qua kết nối dữ liệu. Ví dụ NVT-ASCII có logical byte size 8 bits. Nếu kiểu biểu diễn là local byte , thì TYPE command có một thông số bắt buộc cụ thể thứ hai là logical byte size. Transfer byte size (kích thước byte truyền dữ liệu) luôn là 8 bit.

7.3.1.1.1. ASCII TYPE

- ASCII là kiểu dữ liệu mặc định và nó được chấp nhận bởi tất cả FTP triển khai. Nó có ý định ban đầu là truyền dữ liệu files text, trừ khi các host đã tìm thấy kiểu EBCDIC thuận tiện hơn.

- Sender chuyển dữ liệu từ kí tự biểu diễn bên trong thành chuẩn biểu diễn 8 bit. Receiver sẽ chuyển dữ liệu từ standard form (mẫu chuẩn) về mẫu bên trong của riêng nó.

- Sử dụng chuẩn NVT-ASCII biểu diễn có nghĩa là dữ liệu phải thể hiện dưới dạng 8 bits.

7.3.1.1.2. EBCDIC TYPE

- Loại này được sử dụng để chuyển hiệu quả giữa các máy sử dụng EBCDIC để biểu diễn các kí tự bên trong.

-Dữ liệu biểu diễn 8 bit ký tự EBCDIC . Mã kí tự là sự khác biệt duy nhất giữa các đặc điểm chức năng EBCDIC và ASCII. End-of-line sẽ chắc chắn ít được sử dụng với kiểu EBCDIC cho mục đích biểu diễn cấu trúc, nhưng khi cần thiết kí tự <NL> nên được sử dụng.

7.3.1.1.3. IMAGE TYPE

- dữ liệu được gửi như các bits liên tiếp, dữ liệu được đóng gói vào các byte truyền 8 bit. Phía nhận phải lưu dữ liệu như các bits liên tiếp. Cấu trúc của hệ thống lưu trữ có thể đòi hỏi bộ đệm cho một số giới hạn thuận tiện (byte, word, block). Bộ đệm có thể tất cả ở trạng thái không, có thể xảy ra ở cuối file hoặc cuối mỗi bản ghi và phải có cách xác định các bit bộ đệm để chúng bị tước bỏ khi file lấy ra. Phần đệm chuyển đổi cần được công bố công khai để cho phép người dùng xử lý một tệp tin tại phía lưu trữ.

- Image type được sử dụng cho lưu trữ, truy suất file hiệu quả và cho việc truyền dữ liệu nhị phân (binary data). Nó được đề nghị chấp nhận bởi bởi tất cả FTP triển khai.

7.3.1.1.4 LOCAL TYPE

- Dữ liệu được truyền bằng logical byte có kích thước được xác định bởi tham số bắt buộc thứ hai, bytes size. Giá trị của bytes size dưới dạng số nguyên thập phân, không có giá trị mặc định. Logical byte size không nhất thiết giống tranfer byte size.

- Khi dữ liệu tới receive host, nó được biến đổi trong một kiểu phụ thuộc vào logical byte size và chi tiết host cụ thể.

-Ví dụ user sending 36 bit số thực dấu phẩy động đến host với 32-bits word cũng có thể gửi dữ liệu như local byte với logical byte size là 36. Receive host chờ đợi để lưu trữ logical byte nên có thể dễ dàng thao tác , trong ví dụ đặt 36 bit logical byte vào 64-bits đôi word nên đủ.

7.3.1.1.5 FORMAT CONTROL

- Các Type ASCII và EBCDIC cũng có một thông số thứ 2 tùy chọn, nó chỉ ra loại vertical format control, nếu có, kết hợp với file. Các kiểu dữ liệu biểu diễn sau được định nghĩa trong FTP:

Một character file được truyền đến một host vì một trong 3 mục đích: để in, để lưu trữ và sau đó lấy lại hoặc xử lý. Nếu 1 file được gửi đê in host nhận phải biết cahc vertical format control được biểu diễn. Trong trường hợp thứ hai,có thể lưu trữ 1 file ở một host và sau đó nhận lại file đó ở cùng dạng. Và cuối cùng có thể chuyển từ 1 host đến host khác để xử lý mà không phải lo lắng vấn đề gì. Một định dạng ASCII và EBCDIC thì không thỏa mãn được những điều kiện này, nên những type này cần thông số thứ 2 mô tả một trong 3 format: Non print, Telnet format controls, Carriage Control

7.3.1.2 Cấu trúc dữ liệu (DATA STRUCTURES)

**Phần 8,9,10 …**

**Phần 11 Tổng kết**

**Phần 12 Danh mục tài liệu tham khảo**

1. <https://tools.ietf.org/pdf/rfc1579.pdf>
2. <https://www.ietf.org/rfc/rfc959.txt>
3. <https://tools.ietf.org/pdf/rfc2228.pdf>
4. <https://tools.ietf.org/pdf/rfc2428.pdf>
5. <http://www.networksorcery.com/enp/protocol/ftp.htm>