3.7.3 Truyền dữ liệu

3.7.3.1 Kiểu dữ liệu (Data Types)

- Các đại diện dữ liệu được xử lý trong FTP bởi một người sử dụng chỉ rõ một kiểu đại diện. Kiểu đại diện này có thể ngầm (như trong ASCII hoặc EBCDIC) hoặc rõ ràng (như trong byte địa phương) định nghĩa một kích thước byte cho việc giải thích được gọi là “kích thước byte hợp lý”. Lưu ý rằng điều này không đáng kể với kích thước byte được sử dụng cho sự chuyển giao để hoàn thành kết nối dữ liệu, gọi là “ kích thước byte truyền”.

3.7.3.1.1 Kiểu ASCII (ASCII Type)

- Đây là kiểu mặc định và phải được chấp nhận bởi tất cả các FTP triển khai. Nó chủ yếu dành cho việc chuyển giao các tập tin văn bản, trừ khi cả hai máy chủ tìm thấy kiểu EBCDIC tiện lợi hơn.

- Người gửi chuyển đổi dữ liệu từ một ký tự bên trong đại diện cho NVT-ASCII 8-bit tiêu chuẩn. Người nhận sẽ chuyển đổi dữ liệu từ mẫu tiêu chuẩn sang mẫu nội bộ mình sở hữu.

- Theo tiêu chuẩn NTV, trình tự <CRLF> nên được sử dụng khi cần thiết để chỉ rõ sự kết thúc của một dòng văn bản.

- Sử dụng các tiêu chuẩn NTV-ASCII đại diện có nghĩa là dữ liệu phải được hiểu như là bytes 8-bit.

3.7.3.1.2 Kiểu EBCDIC (EBCDIC Type)

- Kiểu này được dùng để chuyển đổi có hiệu quả giữa các máy chủ có sử dụng EBCDIC cho các ký tự bên trong đại diện.

- Để truyền tải, dữ liệu được biểu diễn dưới dạng kí tự EBCDIC 8-bit. Mã kí tự là sự khác biệt duy nhất giữa các đặc điểm chức năng của EBCDIC và ASCII.

- Kết thúc dòng có lẽ sẽ hiếm khi được sử dụng với kiểu SBCDIC cho các mục đích của cấu trúc biểu diễn, nhưng cần phải sử dụng kí tự ở nơi cần thiết.

3.7.3.1.3 Kiểu hình ảnh (IMAGE Type)

- Dữ liệu được gửi đi dưới dạng các bit liên tiếp, được đóng gói vào các byte truyền 8-bit. Nơi tiếp nhận phải chứa được dữ liệu là các bit liên tiếp. Cấu trúc của hệ thống lưu trữ có thế cần phần đệm của tệp tin (hoặc của bản ghi, cho một tập tin có cấu trúc bản ghi) đến một số ranh giới thuận tiện (byte, word, block).

- Phần đệm chuyển đổi cần được công khai để cho phép người dùng xử lý một tập tin tại nơi lưu trữ.

- Kiểu hình ảnh là dành cho việc lưu trữ có hiệu quả và truy xuất dữ liệu của tệp tin và cho việc chuyển đổi dữ liệu nhị phân. Nó được đề nghị rằng kiểu này được chấp nhận bởi tất cả các FTP triển khai.

3.7.3.1.4 Kiểu địa phương (LOCAL Type)

- Dữ liệu được truyền bằng các byte hợp lý có kích thước quy định bởi tham số thứ hai bắt buộc, kích thước Byte. Giá trị của kích thước Byte phải là một số nguyên thập phân, không có giá trị mặc. Kích thước byte hợp lý không nhất thiết phải giống như kích thước byte truyền. Nếu có sự khác biệt trong kích thước byte thì các byte hợp lý cần đóng gói liên tục, bỏ qua các ranh giới truyền byte và bất kỳ phần đệm nào ở cuối.

- Khi dữ liệu đến máy chủ tiếp nhận, nó sẽ biến đổi theo cách thức phụ thuộc vào kích thước byte hợp lý và máy chủ riêng biệt. Sự biến đổi này cần được công khai bởi các nhà triển khai FTP.

3.7.3.2 Cấu trúc dữ liệu (Data Structures)

* Ngoài các kiểu đại diện khác nhau, FTP cho phép cấu trúc một tệp được chỉ định. Ba tệp cấu trúc được định nghĩa trong FTP là:

+ Cấu trúc tệp: không có cấu trúc bên trong và các tệp được xem như là một chuỗi các byte dữ liệu liên tiếp nhau.

+ Cấu trúc bản ghi: file được tạo thành từ một dãy các bản ghi

+ Cấu trúc trang: file được tạo thành từ các trang có chỉ số độc lập

* Cấu trúc tệp mặc định được giả định nếu lệnh cấu trúc không được sử dụng nhưng cả cấu trúc tệp và cấu trúc bản ghi phải được chấp nhận cho các tệp “văn bản”(các tệp kiểu ASCII và EBCDIC) bởi tất cả các nhà triển khai FTP. Cấu trúc tệp sẽ ảnh hưởng đến cả cách thức chuyển tệp và việc biểu diễn và lưu trữ tệp. Cấu trúc tự nhiên của một tệp sẽ phụ thuộc vào việc lưu trữ nó trên các máy chủ. Một tệp mã nguồn thường được lưu trữ trên một IBM Mainframe trong các bản ghi có chiều dài cố định còn ở trên DEC TOPS-20 là một luồng các kí tự được chia thành dòng. Với một số trang web được định hướng tệp tin tự nhiên và định hướng bản ghi tự nhiên khác có thể có vấn đề nếu một tệp với một cấu trúc được gửi tới một máy chủ định hướng khác. Nếu một tệp văn bản gửi đi với cấu trúc bản ghi đến một máy chủ đó là tệp tin định hướng, sau đó máy chủ đó có thể áp dụng một sự biến đổi nội bộ đến tệp tin dựa trên cấu trúc bản ghi. Trong trường hợp một tệp tin đang được gửi đi với cấu trúc tệp đén một máy chủ định hướng theo bản ghi, có tồn tại vấn đề gì máy chủ nên sử dụng tiêu chuẩn để phân chia các tệp tin vào bản ghi có thể tiến hành cục bộ. Nếu việc phân chia này là cần thiết, việc triển khai FTP có thể sử dụng trình tự kết thúc của dòng, <CRLF> cho ASCII, hoặc <NL> cho các tệp văn bản EBCDIC, như là dấu phân cách. Nếu thực hiện FTP thông qua kỹ thuật này, nó phải được chuẩn bị để đảo ngược sự chuyển đổi nếu tệp tin được lấy ra với cấu trúc tệp.
* Để truyền các tệp tin không liên tục, FTP định nghĩa một cấu trúc trang. Các tệp kiểu này đôi khi được gọi là “các tệp truy cập ngẫu nhiên” hoặc “các tệp có lỗ hổng”. Trong các tệp đó đôi khi có những thông tin khác liên kết với các tệp tin thành tổng thể hoặc với một phần của tệp tin, hoặc cả hai. Trong FTP, các thành phần của tệp tin được gọi là các trang. Để cung cấp các kích thước các trang khác nhau và liên kết thông tin, mỗi trang được gửi cùng với tiêu đề trang. Tiêu đề của trang có các trường được định nghĩa như sau:

+ Độ dài tiêu đề là số byte logic trong tiêu đề của trang. Độ dài tiêu đề thấp nhất là 4.

+ Chỉ số trang là số trang logic phân chia từ tệp. Đây không phải là số thứ tự truyền của trang, nhưng sử dụng chỉ số để nhận biết trang nào của tệp tin.

+ Độ dài dữ liệu là số byte logic của trang dữ liệu. Độ dài dữ liệu thấp nhất là 0

+ Kiểu trang: có các kiểu trang được định nghĩa như sau:

0 = trang cuối: chỉ ra trang cuối cùng của một quá trình chuyển cấu trúc trang. Độ dài tiêu đề là 4, độ dài dữ liệu là 0.

1 = trang đơn: đây là kểu thông thường cho các trang tệp tin đơn giản, không có cấp trang kết hợp thông tin điều khiển. Độ dài tiêu đề là 4.

2 = trang miêu tả: đây là kiểu trang sử dụng để truyền thông tin mô tả cho các tệp tin nói chung.

1. = trang kiểm soát truy cập: trang này bổ sung môi trường tiêu đề cho các trang với cấp độ trang truy cập điều khiển thông tin. Độ dài tiêu đề là 5.

* Các trường tùy chọn: Các trường tiêu đề bổ sung có thể được sử dụng để cung cấp trên mỗi trang điểu khiển thông tin, ví dụ, trên mỗi trang truy cập điều khiển.
* Tất cả các trường là một byte logic về chiều dài. Kích thước byte logic được xác định bởi lệnh TYPE
* Một chú ý về các thông số: một tệp tin phải được lưu trữ và lấy ra với các tham số tương tự nếu phiên bản đã lấy giống hệt với phiên bản truyền ban đầu. Ngược lại, việc thực hiện FTP phải trả về một tệp giống với bản gốc nếu các tham số được sử dụng để lưu trữ và khôi phục một tệp tin là như nhau.

3.7.3.3 Phương thức truyền (Transmission Modes)

* Có 3 phương thức truyền: một phương thức định dạng dữ liệu và cho phép khởi động lại phương thức; một phương thức nén dữ liệu để truyền có hiệu quả; và một phương thức đi qua dữ liệu được xử lý một chút hoặc không. Trong phương thức nén, kiểu đại diện xác định các byte bổ sung.
* Tất cả các dữ liệu truyền phải được hoàn thành với một tệp tin kết thúc (EOF) có thể được định rõ hoặc ngụ ý bởi sự kết thúc của kết nối dữ liệu. Đối với các tệp có cấu trúc bản ghi, điểm kết thúc bản ghi là rõ ràng, bao gồm cả điểm cuối cùng. Đối với các tệp được truyền trong cấu trúc trang một “trang cuối” là kiểu trang đã sử dụng.
* Các phương thức truyền sau đây được định nghĩa trong FTP:
  + - * 1. Phương thức luồng (Stream Mode)

Dữ liệu được truyền đi dưới dạng một luồng byte, ở đó không hạn chế sử dụng kiểu đại diện; cấu trúc bản ghi được thừa nhận.

Trong một tệp cấu trúc bản ghi EOR và EOF sẽ được chỉ định bởi một mã điều khiển hai byte. Byte đầu tiên của mã điều khiển tất cả đều là 1, kí tự Escape. Byte thứ hai sẽ có bit lệnh thấp và các số 0 cho EOR và bit lệnh thấp thứ hai cho EOF; tức là byte sẽ có giá trị bằng 1 cho EOR và giá trị 2 cho EOF. EOR và EOF có thể được chỉ ra đồng thời trên byte cuối cùng truyền bằng cách đổi cả hai bit thấp trên (nghĩa là giá trị 3). Nếu một byte gồm tất cả số 1 dự định được gửi dưới dạng dữ liệu, nó cần được lặp lại trong byte thứ hai của mã điều khiển.

Nếu cấu trúc là một cấu trúc tệp, EOF được chỉ định bởi các máy chủ gửi đóng kết nối dữ liệu và tất cả các byte là byte dữ liệu.

* + - * 1. Phương thức khối (Block Mode)

Tệp được truyền đi dưới dạng một dãy các khối dữ liệu có trước một hoặc nhiều byte tiêu đề. Các byte tiêu đề bao gồm một trường đếm và mã kí hiệu. Trường đếm cho biết tổng độ dài dữ liệu của khối dữ liệu theo byte, từ đó đánh dấu bắt đầu khối dữ liệu tiếp theo (không có các bit bổ sung). Mã kí hiệu chỉ rõ: khối cuối cùng trong tệp (EOF) khối cuối cùng trong bản ghi (EOR), khởi động lại đánh dấu hoặc nghi ngờ dữ liệu (tức là, dữ liệu đang truyền bị nghi ngờ là lỗi và không đáng tin cậy). Mã cuối cùng này không dùng để kiểm soát lỗi trong FTP. Nó được thúc đẩy bởi sự mong muốn của cá trang web trao đổi một số kiểu dữ liệu (ví dụ, dữ liệu địa chấn hoặc thời tiết) để gửi và nhận tất cả dữ liệu cho dù lỗi địa phương (chẳng hạn như băng từ đọc lỗi). Các cấu trúc bản ghi được chấp nhận trong phương thức này, và bất kỳ kiểu đại diện nào cũng có thể được sử dụng.

Tiêu đề gồm có 3 byte. Trong 24 bit của thông tin tiêu đề, 16 bit bậc thấp đại diện cho byte đếm, và 8 bit bậc cao đại diện cho mã mô tả như hình dưới đây:

Khối tiêu đề

+----------------+----------------+----------------+

| Descriptor | Byte Count |

| 8 bits | 16 bits |

+----------------+----------------+----------------+

Mã mô tả được biểu thị bởi các cờ bit trong byte mô tả. Bốn mã được chỉ định, trong đó mỗi mã là giá trị thập phân của bit tương ứng trong byte

Mã Ý nghĩa

128 Cuối khối dữ liệu là EOR

64 Cuối khối dữ liệu là EOF

32 Các lỗi nghi ngờ trong khối dữ liệu

16 Khối dữ liệu là một dấu hiệu khởi động

Với mã hóa này, nhiều hơn một mã mô tả điều kiện có thể tồn tại cho một khối cụ thể. Có nhiều bit khi cần thiết có thể bị gắn cờ.

Dấu hiệu khởi động lại được nhúng trong luồng dữ liệu là một số nguyên của byte đại diện 8 bit có thể in được kí tự trong ngôn ngữ hiện tại được sử dụng trong kiểm soát kết nối .

* + - * 1. Phương thức nén (Compressed Mode)

Có 3 loại thông tin được gửi: dữ liệu chính quy được gửi bằng một chuỗi byte; dữ liệu nén bao gồm bản sao chép hoặc bổ sung; và thông tin điều khiển được gửi bằng trình tự thoát 2 byte. Nếu n > 0 byte (đến 127) dữ liệu chính quy được gửi, n byte này được đặt trước bởi 1 byte với bit trái nhất đặt bằng 0 và 7 bit phải nhất có chứa số n.

Chuỗi byte:

1 7 8 8

+-+-+-+-+-+-+-+-+ +-+-+-+-+-+-+-+-+ +-+-+-+-+-+-+-+-+

|0| n | | d(1) | ... | d(n) |

+-+-+-+-+-+-+-+-+ +-+-+-+-+-+-+-+-+ +-+-+-+-+-+-+-+-+

^ ^

|---n bytes---|

Dữ liệu

Chuỗi n byte dữ liệu d(1), ... , d(n)

Với n dương

Để nén một chuỗi của n bản sao chép dữ liệu byte d, sau 2 byte được gửi đi:

2 6 8

+-+-+-+-+-+-+-+-+ +-+-+-+-+-+-+-+-+

|1 0| n | | d |

+-+-+-+-+-+-+-+-+ +-+-+-+-+-+-+-+-+

Một chuỗi của n byte bổ sung có thể được nén thành một byte đơn, nơi mà các byte bổ sung biến đổi với các kiểu đại diện. Nếu kiểu là ASCII hoặc EBCDIC byte bổ sung là <SP> (Space, mã ASCII 32, mã EBCDIC 64). Nếu kiểu là hình ảnh hoặc byte cục bộ byte bổ sung là một byte 0.

2 6

+-+-+-+-+-+-+-+-+

|1 1| n |

+-+-+-+-+-+-+-+-+

Trình tự thoát là một byte kép, đầu tiên là byte thoát (tất cả là 0) và byte thứ hai chứa đựng mã mô tả như định nghĩa trong Phương thức Khối. Mã mô tả có ý nghĩa giống như ở phương thức Khối và áp dụng cho chuỗi kế tiếp của byte.

Phương thức nén là hữu ích

3.7.3.4 Thiết lập kênh dữ liệu

3.7.3.4.1 Phương thức chủ động (Active Mode)

* Tạo kết nối kiểm soát: qua giao thức bắt tay ba bước

+ FTP Client mở một cổng cao bất kỳ gửi gói tin SYN với mục đích thông báo yêu cầu kết nối với FTP Server

+ FTP Server đồng ý và gửi lại gói tin SYN/ACK

+ FTP Client nhận SYN/ACK và gửi lại gói tin ACK, đồng ý tạo kết nối . Tại thời điểm này kết nối được thiết lập.

* Tạo kết nối dữ liệu:

+ FTP Client: sử dụng kết nối được tạo trước đó để gửi lệnh PORT yêu cầu FTP server dùng phương thức chủ động để truyền file đồng thời thông báo cho FTP server biết nó sẽ mở một cổng mới để chờ tạo kết nối dữ liệu

+ FTP Server: nhận được yêu cầu và tiến hành bắt tay ba bước với FTP Client.

Quá trình tạo kết nối dữ liệu diễn ra rất nhanh. Sau khi hoàn thành FTP Server sẽ gửi dữ liệu về cho FTP Client.

Tóm lại cách thức hoạt động của FTP theo phương thức chủ động là: phía FTP Client khởi tạo kiểm soát kết nối, còn phía FTP Server khởi tạo kết nối dữ liệu. FTP Server là bên chủ động gửi dữ liệu cho FTP Client.

3.7.3.4.2 Phương thức bị động (Passive Mode)

* Tạo kết nối kiểm soát: theo quy tắc bắt tay ba bước như phương thức chủ động

+ FTP Client: gửi gói tin SYN yêu cầu kết nối

+ FTP Server: nhận gói tin SYN, đồng ý và gửi lại gói tin SYN/ACK

+ FTP Client: nhận gói tin SYN/ẠCK và gửi lại gói tin ACK để thông báo. Kết nối được thiết lập

* Tạo kết nối dữ liệu:

+ FTP Client: chủ động gửi lệnh PASV FTP Server để có thể truyền dữ liệu theo phương thức bị động.

+ FTP Server: nhận được yêu cầu cần phải dùng phương thức bị động để truyền dữ liệu từ FTP Client, nó trả lời lại cho FTP Client biết sẽ dùng cổng nào để tạo kết nối dữ liệu.

+ FTP Client: nhận được phản hồi từ FTP Server nó sễ mở một cổng để tiến hành giao thức bắt tay ba bước với FTP Server (như kết nối kiểm soát). Sau khi hoàn thành, lúc này kết nối dữ liệu được thiết lập.

* FTP Client muốn nhận dữ liệu chỉ việc gửi thông điệp đến FTP Server qua kết nối điều khiển. FTP Server sẽ truyề dữ liệu cho FTP Client qua kết nối dữ liệu.

Tóm lại trong phương thức bị động FTP Client khởi tạo cả hai kết nối đến FTP Server. FTP Client là phía yêu cầu nhận dữ liệu.

3.7.3.4.1 Tường lửa

* Giao thức FTP sử dụng kết nối TCP thứ cấp trong thực tế truyền tải các tệp tin. Theo mặc định, kết nối này được thiết lập bởi một phương thức mở từ FTP Server đến FTP Client. Tuy nhiên, điều này không hoạt động tốt với tường lửa dựa trên bộ lọc gói.
* Mặt khác, kmáy khách sử dụng lệnh PASV, dữ liệu kênh sẽ là một cuộc gọi đi qua tường lửa. Các cuộc gọi như vậy dễ dàng xử lý hơn và ít gặp vấn đề hơn.
* Theo mặc định, tất cả các dữ liệu truyền nên được hoàn thành trên một kết nối duy nhất. Một chủ động mở thực hiện bởi máy chủ, từ cổng 20 đến cùng một cổng trên máy khách như đã được sử dụng cho kết nối điều khiển. Máy khách làm một bị động mở.
* Hầu hết các máy khách hiện hành không làm theo cách đó. Một kết nối mới được sử dụng cho mỗi lần truyền; để tránh chạy vào trạng thái thời gian chờ (TIMEWAIT) của TCP, máy khách chọn một cổng mới mỗi lần và gửi một lệnh PORT thông báo đến máy chủ.
* Nếu bộ lọc gói được sử dụng (được cung cấp bởi hầu hết các bộ định tuyến hiện đại), kênh dữ liệu yêu cầu xuất hiện như cuộc gọi đến các cổng không xác định. Hầu hết các tường lửa được xây dựng cho phép các cuộc gọi đến các cổng xác thực độ an toàn tin cậy, chẳng hạn như SMTP. Thông thường sự thỏa hiệp là để ngăn chặn khu vực máy chủ, nghĩa là số cổng dưới 1024. Nhưng chiến lược này đầy rủi ro, các dịch vụ nguy hiểm như X Windows sống ở các cổng cao hơn.
* Các cuộc gọi đi gặp ít vấn đề hơn, hoặc cho quản trị viên tường lửa hoặc cho bộ lọc gói. Bất kì gói tin TCP nào với bộ bit ACK không thể là gói được sử dụng để bắt đầu một kết nối TCP; các bộ lọc có thể được cấu hình để truyền các gói như vậy. Do đó cần thay đổi cách xử lý của FTP để kênh dữ liệu được thực hiện như một cuộc gọi từ máy khách đến máy chủ.
* May mắn là các cơ chế cần thiết đã tồn tại trong giao thức. Nếu máy khách gửi một lệnh PASV, máy chủ sẽ thực hiện TCP dạng bị động mở trên một số cổng ngẫu nhiên, và thông báo cho máy khách về số cổng. Máy khách sau đó có thể thực hiện một hoạt động mở để thiết lập kết nối.
* Với cách xử lý của phần lớn các máy khách FTP hiện nay, việc sử dụng PASV không gây ra bất kỳ tin nhắn bổ sung nào được gửi đi. Trong mọi trường hợp, một hoạt động chuyển tiếp được thực hiện bởi sự trao đổi thêm giữa máy khách và máy chủ; nó không phải vấn đề nếu trao đổi bao gồm một lệnh PORT và một lệnh PASV.
* Sử dụng PASV làm nâng cao sự an toàn của máy cổng, vì học không còn phải tạo ra các cổng mà người ngoài có thể kết nối tới. Quan trọng hơn giao thức giữa máy khách và tường lửa có thể được đơn giản hóa, nếu nó không cần xác định một hoạt động tạo.
* Với việc sử dụng PASV, máy chủ FTP phải chấp nhận các cuộc gọi đến cổng ngẫu nhiên , có thể gây ra vấn đề bất bình đẳng với tường lửa của nó. Nhưng đây không phải là vấn đề nghiêm trọng vì nhiều lý do. Đầu tiên, có rất ít máy chủ FTP so với máy khách. Nó có thể được đảm bảomột số lượng nhỏ các máy móc đặc biệt, chẳng hạn như các cổng và tổ chức các máy chủ FTP. Bộ lọc tường lửa có thể được cấu hình để cho phép truy cập vào các máy này. Hơn nữa, các biện pháp phòng ngừa khác có thể được thực hiện bằng cách sửa đổi máy chủ FTP sao cho nó chỉ sử dụng các cổng rất cao cho kênh dữ liệu. Nó tương đối dễ dàng để đảm bảo rằng không có dịch vụ nguy hiểm nào sống trong dãy cổng.