

# FTP에서 Stream 전송 모드의 개선

박성인<sup>0</sup>, 최돈은, 김동한, 서영상, 서정수, 이재영  
한림대학교 컴퓨터공학과

Improvement of Stream Transmission Mode in File Transfer Protocol

S.I.Park<sup>0</sup>, D.E.Choi, Y.S.Seo, D.H.Kim, J.S.Seo, J.Y.Lee  
Hallym Univ. Department of Computer Engineering

## 요 약

RFC 959에서 정의한 FTP의 세 가지 전송모드, Stream, Block, Compressed 전송 모드 중 현재 주로 사용되고 있는 전송 모드는 Stream 전송 모드인데, 이 Stream 전송 모드에서 서버 또는 네트워크 치명적인 오류 발생하였을 경우 이를 체크 할 수 없고 서버에 재 접속하였을 경우 처음부터 재전송 하여야 하는 문제점이 발생한다. 이를 개선한 Block, Compressed 전송 모드의 경우, 데이터 스트림에 정보 헤더를 추가하여야 함으로써 오버 헤드를 발생한다. 본 논문에서는 클라이언트 FTP상에서 전송 상태를 체크하게 함으로써 전송 중단시 중단된 시점부터 나머지 데이터를 보냄으로써 FTP의 스트림 전송 모드의 문제점을 개선하는 방법을 제시한다.

## 1. 서론

ARPANET 초기부터 TCP/IP의 일부로 존재하며 정보 공유를 그 목적으로 사용되어진 FTP는 1971년 MIT에서 호스트들간의 자료의 공유를 위해서 개발한 파일 전송 메커니즘을 RFC 114에 규정한 이래로 약 25년 동안 40여 회의 수정을 거듭한 후, 현재 사용하는 FTP의 규격으로 1985년에 정의된 RFC 959를 이용하고 있다[2, 3, 4].

FTP에서 사용하는 세 가지 전송 모드로 Stream, Block, Compressed를 규정하고 있다[1, 2]. 이들 중 현재 가장 많이 사용되는 전송 모드는 Stream 전송 모드인데, 이 전송 모드는 자료 전송 중 치명적인 오류 발생 시 전송중인 파일에 대하여 처음부터 재 전송해야만 하는 문제점을 가진다. 이에 대한 개선안으로 Block, Compressed모드가 정의되어 있으나, 현재 많은 FTP서버들이 이를 지원하지 않고 있다. 이를 방법은 각 프레임들에 정보 헤더들이 추가됨으로써 네트워크 서버에 오버 헤드를 발생시킨다[6].

따라서 본 논문에서는 스트림 전송모드에서 데이터 전송 중 전송 중단시, 전송량을 계산하여 중단 시점부터 나머지 데이터를 전송할 수 있도록 서버와 클라이언트 부분을 수정하여 개선한다.

## 2. FTP

### 2.1 FTP 모델

그림 1에서 FTP는 파일의 송수신을 위해서 두 가지 TCP연결, 즉, 제어 연결과 데이터 연결을 이용한다. 이들 중 제어 연결을 이용하여 사용자 FTP는 서버 FTP로 명령을 보내고 응답을 받게되며, 실제의 데이터 파일은 데이터 연결을 이용하여 전송한다.

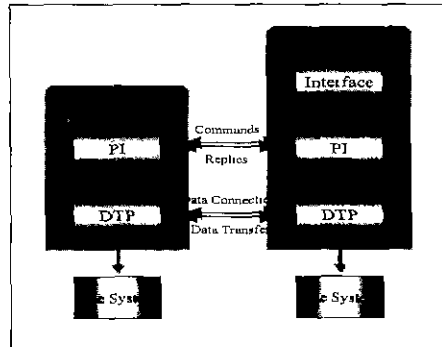


그림 1. FTP 모델

사용자 FTP는 사용자 인터페이스, 사용자 PI(Protocol Interpreter), 사용자 DTP(Data Transfer Process)의 세 가지 논리적 모듈로

구성되고, 서버 FTP는 서버 PI, 서버 DTP 두 가지 논리적 모듈로 구성된다.

사용자 인터페이스는 사용자와 대화하기 위한 인터페이스를 제공하며, 사용자와 서버의 PI는 FTP프로토콜을 해석한 후 각각의 DTP를 제어하는 역할을 한다.

사용자와 서버의 DTP는 PI의 제어에 따라 파일을 관리 또는 전송한다. 즉, PI는 전송이 시작될 때와 전송 방향, 그리고 교환될 데이터의 포맷을 DTP에게 알려 주며, DTP는 데이터 전송이 완료되거나 에러가 발생하였을 때 이를 PI에게 알려준다.

### 2.2 FTP의 연결과 자료전송

FTP에서 사용자 FTP는 서버와 접속을 하게 되면 먼저 제어 연결을 시도 후 TCP 포트 21번을 사용하여 접속이 이루어진다. 이때 제어 연결은 Telnet Protocol을 이용하여 이루어지게 된다. 일단 제어 연결을 이용하여 사용자에게 대한 인증이 이루어지면 사용자는 FTP에서 제공하는 여러 가지 명령들을 이용할 수 있다. 명령과 응답은 7비트의 ASCII테스트로 전송이 이루어지며 CRLF로 종료되어진다

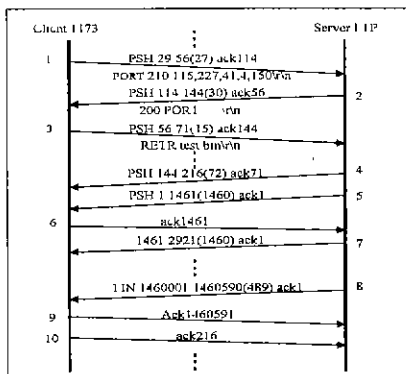


그림2. FTP에서의 자료전송

실제적인 자료의 전송은 그림 2와 같이 사용자와 서버의 PI들간의 명령과 응답이 이루어진 후, PI의 제어에 의해 양측 DTP 간에 PORT 20 또는 22 번을 이용하여 자료의 전송이 이루어진다. DTP 간에 자료의 전송이 완료된 다음, 전송 모드가 Stream 전송 모드이면 서버 쪽에서는 연결을 닫음으로써 사용자 FTP에 EOF임을 나타낸다.

### 2.3 세 가지 전송 모드

FTP에서 데이터의 포맷과 구조가 서버와 사용자 FTP간의 PI정의를 이루어지면, 양측은 반드시 세 가지 전송 모드, Stream, Block, Compressed 중에 하나의 전송 모드를 결정해야만 한다.

디폴트 전송모드인 Stream모드는 모든 FTP에서 사용되고 있으나 세 가지의 모드 중 가장 신뢰성이 낮다. Stream전송모드는 데이터 스트림에서 차례에 대한 정보를 갖지 않으므로 사용자측은 EOF지점이 도착했는지를 구분할 수 없으며, 오류에 대한 검사는 TCP수준에서 처리되므로 치명적인 오류에 대하여 대처 할 수 없다.

반면, Block과 Compressed모드는 사용자로부터 EOF가 도착했는지를 알 수 있는 정보를 판단할 수 있게 하기 위한 차례 정보를 헤더 형태로 가지고있다. Block, Compressed 모드는 파일이 전송된 다음 데이터의 연결을 닫을 것을 요구하지 않으며 파일 전송이 중단되었을 경우 전송이 중단된 다음부터 재전송이 가능하나 오버헤드가 추가된다 또한 현재 대부분의 서버의 경우 이들 모드를 지원하지 않고 있으며 때문에 대부분의 FTP 클라이언트 또한 이 모드를 사용하지 않고 있다.

### 3 스트림 전송모드의 개선 방법

#### 3.1 스트림 전송 모드에서의 문제점

Stream 전송 모드의 경우, 단순히 파일을 바이트의 연속으로 전송하며 데이터가 차례대로 수신되도록 보장하는 TCP의 여러 제어 장치를 이용한다. 따라서 전송 중 서버 또는 네트워크 상에 치명적인 오류가 발생되면 모든 데이터가 전송되기 전에 연결이 닫히게 된다. 이 때문에는 잘못된 파일 전송이 이루어 졌다는 사실을 알 수 없게 되며 알게 되더라도 파일을 처음부터 다시 전송해야만 하므로 대용량의 데이터의 경우 많은 재 전송시간이 요구되어진다. 그림 3은 치명적인 오류로 인한 데이터 전송의 단절을 보여주고 있다

반면, Block, Compressed 전송 모드의 경우 Stream모드의 단점을 개선하기는 하였지만 이를 위해 매 스트림마다 Block 전송 모드의 경우 1byte이상, Compressed 전송 모드의 경우 2byte의 헤더 정보를 기록하여 보내야

만 하므로 오버 헤드가 발생한다

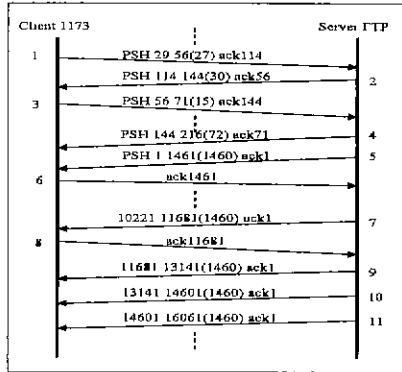


그림 3. 치명적 오류에 의한 FTP 단절

### 3.2 스트림 전송 모드의 개선안

아직까지도 FTP는 많이 사용되고 있는 프로토콜중 하나이며 이는 멀티미디어 데이터와 같은 대용량의 데이터 전송 시 더욱 유용하다. 그러나 그림 3과 같이 서버 또는 네트워크 상태의 치명적 오류 발생 시 3.1에서와 같은 문제점들로 인하여 마지막 지점부터의 재전송이 불가능하므로 이에 따르는 많은 재전송 비용이 들게 된다.

현재 사용중인 대부분의 운영체제는 UNIX와 마이크로 소프트웨어의 Windows등이 대부분을 차지하며 이중 대부분 UNIX간의 파일 크기에는 차이가 없으나, UNIX와 Windows간의 파일의 크기의 차이가 존재하게 된다. 그러나 이러한 파일간의 크기는 Binary상태의 경우 동일하지만, Text의 경우에 UNIX에 비해 Windows의 파일이 매 라인마다 CR 코드가 추가되어 크기는 다르다. 그 차이는 단지 CR의 추가이기 때문에 간단히 UNIX와 Windows간의 Text파일의 크기는 계산할 수 있다.

이를 이용하면 파일의 전송 전에 LIST명령을 서버 FTP보내 그 응답을 받음으로서 전송될 파일의 크기를 알 수 있다. 따라서 사용자 FTP에서는 전송된 데이터 스트림의 크기와 이미 알고 있는 파일의 크기와 비교하여 전송이 정상적으로 이루어졌는지를 알 수 있다. 만일 전송이 치명적인 오류로 인해 중단되었다면 이를 LOG 형태로 기록하게 되고 다음 빈 접속에서 중단된 다음 지점을 서버 FTP에게 알려 줌으로써 중단된 이후의 데이터부터 전송이 이루어지게 된다.

그림 4는 위의 방법을 이용하여 재전송이

이루어지는 과정을 나타내고 있다.

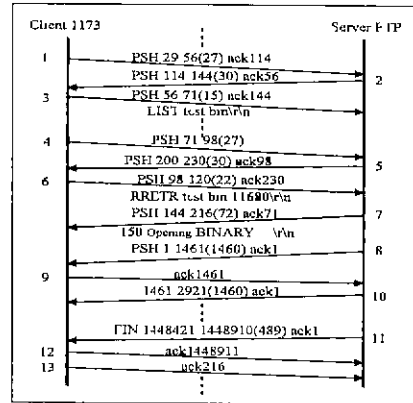


그림 4. 개선안에 따른 파일의 재전송

### 4. 결론

RFC 959에서 정의한 FTP전송 모드의 경우 Stream 전송 모드의 경우 파일의 전송 시 치명적 오류 발생으로 인해 잘못된 데이터의 전송이 이루어지게 되며, Block, Compressed 전송 모드의 경우에도 Stream 전송 모드의 단점을 개선하기는 하였으나 추가 정보 헤더로 인한 오버 헤드가 발생하였다. 그러나 개선안에 따른 FTP에서는 전송 중 치명적 오류 발생 시 이에 대한 검출과 재전송이 가능하고 클라이언트측에서의 간단한 오류 체크로 부가적인 정보 헤더가 불필요하여 Block, Compressed전송 모드에서의 오버 헤드를 없앨 수 있는 장점이 있다.

### 참고문헌

- [1] 최진부, "파일 전송 규약(FTP) 표준", 대한민국 전산망 표준, KIS-3I-0013(93) 1993
- [2] J. Postel and J. Reynolds, "File Transfer Protocol" RFC 950, ISI, October, 1985
- [3] Bhushan, Abhay, "A File Transfer Protocol", RFC 114(NIC 5823), MIT-Project MAC, 16 April 1971
- [4] Harslem, Eric, and John Heifner, "Comments on RFC 114(A File Transfer Protocol)", RFC 141 (NIC6726), RAND, 28 April 1971
- [5] 이 국발생 표준 MIL-STD-1780
- [6] Dave Roberts, Developing for the Internet with Winsock, Corolis Group Books, 1995
- [7] 서정수, 박성민, 김희준, 이용수 최영철 이세영 "문서의 수정 기능을 갖는 File Transfer Protocol" conference on 24th KISS spring, 1997