GWD-R
GridFTP WG
http://www-isd.fnal.gov/gridftp-wg/

Igor Mandrichenko, FNAL

August 29, 2003
Revised August 29, 2003

End of File Communication in Stream Mode

Igor Mandrichenko
August 29, 2003

Introduction

Stream file transfer mode is the simplest and most popular data transfer mode of in FTP and GridFTP protocol. Its simplicity makes it easy to implement and on the other hand makes it less robust with respect to network and other failures. In particular, RFC959 states that in Stream mode, receiving server is supposed to treat disconnection of data socket as the end of the file. This definition makes it impossible for the server to distinguish between successful completion of the transfer and termination of the transfer due to abnormal client shutdown. Here is what happens when uploading client terminates in the middle of file transfer:

```
Client                  Server
STOR file
                        120 Opening data connection
(sending data)          (receiving data)
(client is terminated)
                        (end-of-file on data socket received)
                        226 Data transfer complete
(control socket is closed)
```

As you can see, from the server side, this may be treated as a successful transfer followed by ungraceful client disconnection. There is no way for the server to know that in fact the transfer was incomplete.

This document proposes one of the solutions for this problem.

EOF Command

New optional FTP command “EOF” can be used to make the difference between abnormal and successful file transfer clear. If the server implements EOF command, and its usage is turned on (see section on OPTS usage below), then the client is *required* to issue EOF command after it closes data channel at the end of successful file transfer. Server would always reply with 22x status to confirm its receipt.

The following is the command-reply sequence for normal transfer:

```
Client                  Server
STOR file
                        120 Opening data connection
(sending data)          (receiving data)
                        226 Data transfer complete
EOF
                        227 OK
```

In this case the server receives the EOF command and considers the transfer finished successfully.

Here is what happens if the client terminates in the middle of transfer

```
Client                  Server
STOR file
                        120 Opening data connection
(sending data)          (receiving data)
(client is terminated)
                        (end-of-file on data socket received)
                        226 Data transfer complete
(control socket is closed)
                        (no EOF command is received)
                        (control socket closes before
                        EOF command is received)
```

Because the server never receives EOF command, it considers the transfer to be incomplete and may choose to discard the portion of the file it has received or keep it for further re-transmission.

OPTS Usage

In order to preserve backward compatibility with RFC959, OPTS command can be used to negotiate use of this feature. Client should use OPTS command to query whether the server implements EOF command and to request using this feature:

Command	Meaning
OPTS STOR EOF ON	Request using mandatory EOF command
OPTS STOR EOF OFF	Turn off using EOF command

Possible server replies:

Reply	Meaning
451 …	EOF option is unavailable
501 …	EOF option is not implemented
5xx …	OPTS or EOF are unknown
200 …	EOF option is tuned on or off as requested by the client

By default this feature will be turned off and the server will not require EOF to be sent by the client at the end of the transfer.

If the server refuses to turn the feature on, the client should use RFC959 protocol specifications.